MAN697 - TERM PROJECT

“Software Module for Forecasting Time Series by Trend and Seasonality Corrected Exponential Method(Holt-Winter Model)”

Erdem Yıldırımer

January 2012

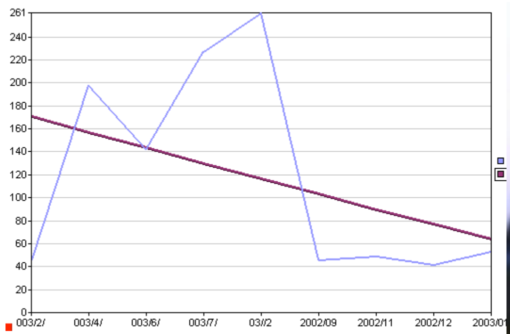

TABLE OF CONTENTS

MAN 697 TERM PROJECT REPORT

"Forecasting Time Series using Trend and Seasonality Corrected Exponential Smoothing Method Software Module"

Erdem Yıldırımer

January 2012

1.Introduction

Exponential smoothing is a simple technique used to smooth and forecast a time series without the necessity of fitting a parametric model. It is based on a recursive computing scheme, where the forecasts are updated for each new incoming observation. Exponential smoothing is sometimes considered as a naive prediction method. . The Holt-Winters method, also referred to as double exponential smoothing, is an extension of exponential smoothing designed for trended and seasonal time series. Holt-Winters smoothing is a widely used tool for forecasting business data that contain seasonality, changing trends and seasonal correlation.

The exponential and Holt-Winters techniques are sensitive to usual events or outliers. Outliers affect the forecasting methods in two ways. First, the smoothed values are affected since they depend on the current and past values of the series including the outliers. The second effect of outliers involves the selection of the parameters used in the recursive updating scheme. These parameters regulate the degree of smoothing and are chosen to minimize the sum of squared forecast errors.

2.Background

Beyond static model mainly adaptive models are explained.

2.1.Time Series Forecasting

In statistics, signal processing ,econometrics and mathematical finance, a time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals. Examples of time series are the daily closing value of the Dow Jones index or the annual flow volume of the Nile River at Aswan. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

2.2.Simple Moving Average

In financial applications a simple moving average is the unweighted meanof the previous n data points. However, in science and engineering the mean is normally taken from an equal number of data either side of a central value. This ensures that variations in the mean are aligned with the variations in the data rather than being shifted in time. of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

2.3. What Is Exponential smoothing?

Exponential smoothing is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make forecasts. The time series data themselves are a sequence of observations. The observed phenomenon may be an essentially random process, or it may be an orderly process. Whereas in the simple moving average the past observations are weighted equally, exponential smoothing assigns exponentially decreasing weights over time.

2.4. Simple Exponential smoothing

Exponential smoothing was first suggested by Charles C. Holt in 1957, although the formulation below, which is the one commonly used, is attributed to Brown and is known as "Brown's simple exponential smoothing".

The simplest form of exponential smoothing is given by the formulae:

$$s_1 = x_0$$
$$s_t = \alpha x_{t-1} + (1-\alpha)s_{t-1} = s_{t-1} + \alpha(x_{t-1} - s_{t-1}), t > 1$$

where α is the *smoothing factor*, and $0 < \alpha < 1$. In other words, the smoothed statistic s_t is a simple weighted average of the previous observation x_{t-1} and the previous smoothed statistic s_{t-1}. The term *smoothing factor* applied to α here is something of a misnomer, as larger values of α actually reduce the level of smoothing, and in the limiting case with $\alpha = 1$ the output series is just the same as the original series (with lag of one time unit). Simple exponential smoothing is easily applied, and it produces a smoothed statistic as soon as two observations are available.

2.5. Double Exponential smoothing

Simple exponential smoothing does not do well when there is a trend in the data. In such situations, several methods were devised under the name "double exponential smoothing".

One method, sometimes referred to as "Holt-Winters double exponential smoothing" works as follows:

Again, the raw data sequence of observations is represented by $\{x_t\}$, beginning at time $t = 0$. We use $\{s_t\}$ to represent the smoothed value for time t, and $\{b_t\}$ is our best estimate of the trend at time t. The output of the algorithm is now written as F_{t+m}, an estimate of the value of x at time $t+m$, $m>0$ based on the raw data up to time t. Double exponential smoothing is given by the formula :

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1-\alpha)(s_{t-1} + b_{t-1})$$
$$b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1}$$
$$F_{t+m} = s_t + mb_t,$$

where α is the *data smoothing factor*, $0 < \alpha < 1$, β is the *trend smoothing factor*, $0 < \beta < 1$, and b_0 is taken as $(x_{n-1} - x_0)/(n-1)$ for some $n>1$. Note that F_0 is undefined (there is no estimation for time 0), and according to the definition $F_1=s_0+b_0$, which is well defined, thus further values can be evaluated.

2.6. Triple Exponential smoothing

Triple exponential smoothing takes into account seasonal changes as well as trends. It was first suggested by Holt's student, Peter Winters, in 1960.

The sequence of observations is again represented by $\{x_t\}$, beginning at time $t = 0$. $\{s_t\}$ represents the smoothed value of the constant part for time t. $\{b_t\}$ represents the sequence of best estimates of the linear trend that are superimposed on the seasonal changes. $\{c_t\}$ is the sequence of seasonal correction factors for time t. L is the period of time of one cycle of seasonal change. The output of the algorithm is again written as F_{t+m}, an estimate of the value of x at time $t+m$, $m>0$ based on the raw data up to time t. Triple exponential smoothing is given by the formulas:

$$
\begin{aligned}
s_0 &= x_0 \\
s_t &= \alpha \frac{x_t}{c_{t-L}} + (1-\alpha) F_t \\
b_t &= \beta (s_t - s_{t-1}) + (1-\beta) b_{t-1} \\
c_t &= \gamma \frac{x_t}{s_t} + (1-\gamma) c_{t-L} \\
F_{t+m} &= (s_t + m b_t) c_{(t+m)} \pmod{L},
\end{aligned}
$$

3.Conclusion

The aim on this study was to create a software module for calculating forecast values.For this Processing Code IDE used. Software accepts 12 user input values from list.txt file and runs on triple exponential method and results are saved to results.txt file. For current version period ,α β and γ values can only be changed within the software.

3.1.Pseudo Code

Given a timse series ,say a complete monthly data for 12 months the Holt-Winters smoothing and forecasting technique is built on following formula :

$$Level: L_t = \alpha \frac{y_t}{S_{t-c}} + (1-\alpha)(L_{t-1})$$

$$Trend: T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

$$Seasonal: S_t = \gamma \frac{y_t}{L_t} + (1-\gamma) * S_{t-c}$$

And predicted smoothed values are computed using the formula:

$$f_{t+m} = (L_t + T_{t+m})S_{t-c}$$

3.2.Processing Code

```
size(200, 200);
int period = 4;
float alfa = 0.1;
float beta = 0.2;
float gamma = 0.1;
String lines[ ] = loadStrings("list.txt");
println("there are " + lines.length + " lines");
int demandLength = lines.length;
float[ ] demand = new float[demandLength];
for (int i=0; i < lines.length; i++) {
  demand[i]=float(lines[i]);
  println(lines[i]);
}
float[ ] indeksler = new float[demandLength];
float[ ] series = new float[demandLength+period];
float[ ] Fseries = new float[demandLength+period];
float[ ] forecast = new float[period];
println (demandLength);
float periodf=float(period);
println(periodf);
float c = demandLength % periodf;
float period1= (demand[0]+demand[1]+demand[2]+demand[3])/periodf;
```

```
println("period1 ortlama="+period1);
float period2= (demand[4]+demand[5]+demand[6]+demand[7])/periodf;
println("period2 ortalama="+period2);
float b0=(period2-period1)/periodf;
println("b0 initial trend = "+b0);
float a=0;
float b=1;
for(int i=1;i<period+1;i=i+1){a=b+a; b=b+1;}
println("a="+a);
float tbar= a/periodf;
println("tbar="+tbar);
float a0 = period1-(b0*tbar);
println("a0="+a0);
for(int i=0;i<demandLength; i=i+1)
  {
    indeksler[i]= demand[i] / (a0+(i+1)*b0) ;
    println("index "+i+" "+indeksler[i]);
    println("series "+i+" "+series[i]);
}
for(int i=0;i<period; i=i+1)
  {   series[i] = (indeksler[i]+indeksler[i+period])/2;
      println("series "+i+" "+series[i]);
  }
float S=0;
for(int i=0;i<period; i=i+1){ S = series[i] +S;}
println("S = "+ S);
float tS = period / S;
println("tS = "+ tS);
for(int i=0;i<period; i=i+1) {
  series[i] = series[i] * tS;
    println("series "+i+" "+series[i]);
  }
float At = a0;
println("At = "+ At);
float Bt = b0;
println("Bt = "+ Bt);
for(int i = 0 ; i<demandLength ; i=i+1)
  { float Atm1=At;
    float Btm1=Bt;
    At = alfa * demand[i]/series[i] + (1.0 - alfa) * (Atm1 + Btm1);
    Bt = beta * (At - Atm1) + (1- beta) * Btm1;
    series[i+period] =gamma * demand[i] / At + (1.0 - gamma) * series[i];
```

```
   Fseries[i]=(a0 + b0 * (i+1)) * series[i];
   print ("i="+ (i+1) +"    ");
   print ("demand="+ demand[i] +"    ") ;
   print ("S=" + series[i] +"    ") ;
   print ("Atm1="+ Atm1 +"    ") ;
   print ("Btm1=" + Btm1 +"    ") ;
   print ("At=" + At + "    ") ;
   print ("Bt=" + Bt + "    ") ;
   print ("series=" + (i+period) + series[i] + "    ");
   println ("Fseries="+ Fseries[i]);
 }
int[ ] intforecast = new int[period];
String[ ] stforecast = new String[period];
for(int i = 0 ; i< period ; i=i+1)
 {
  forecastast[i]=(At + Bt* (i+1))* series[demandLength + i];
  intforecast[i] = (int) forecast[i];
  stforecast[i]= Integer.toString(intforecast[i]);
  println ("Forecast = "+ stforecast[i]);
  saveStrings("result.txt", stforecast);
 }
```

3.3. Example Inputs(list.txt)

9000

13000

23000

24000

10000

18000

23000

38000

12000

13000

32000

41000

3.4. Outputs(result.txt)

15165

23007

35236

45063

3.5. Outputs Within Software

THERE ARE 12 LINES

9000

13000

23000

24000

10000

18000

23000

38000

12000

13000

32000

41000

12

4.0

PERIOD1 ORTLAMA=17250.0

PERIOD2 ORTALAMA=22250.0

B0 INITIAL TREND = 1250.0

A=10.0

TBAR=2.5

A0=14125.0

INDEX 0 0.58536583

SERIES 0 0.0

INDEX 1 0.7819549

SERIES 1 0.0

INDEX 2 1.2867132

SERIES 2 0.0

INDEX 3 1.254902

SERIES 3 0.0

INDEX 4 0.49079755

SERIES 4 0.0

INDEX 5 0.8323699

SERIES 5 0.0

INDEX 6 1.0054644

SERIES 6 0.0

INDEX 7 1.5751295

SERIES 7 0.0

INDEX 8 0.4729064

SERIES 8 0.0

INDEX 9 0.48826292

SERIES 9 0.0

INDEX 10 1.1479821

SERIES 10 0.0

INDEX 11 1.4077253

SERIES 11 0.0

SERIES 0 0.5380817

SERIES 1 0.8071624

SERIES 2 1.1460888

SERIES 3 1.4150157

S = 3.9063487

TS = 1.0239742

SERIES 0 0.55098176

SERIES 1 0.82651347

SERIES 2 1.1735654

SERIES 3 1.4489396

AT = 14125.0

BT = 1250.0

I=1 DEMAND=9000.0 S=0.55098176 ATM1=14125.0
BTM1=1250.0 AT=15470.948 BT=1269.1897
SERIES=40.55098176 FSERIES=8471.345

I=2 DEMAND=13000.0 S=0.82651347 ATM1=15470.948
BTM1=1269.1897 AT=16638.996 BT=1248.9613
SERIES=50.82651347 FSERIES=13740.786

I=3 DEMAND=23000.0 S=1.1735654 ATM1=16638.996
BTM1=1248.9613 AT=18059.0 BT=1283.1699
SERIES=61.1735654 FSERIES=20977.48

I=4 DEMAND=24000.0 S=1.4489396 ATM1=18059.0
BTM1=1283.1699 AT=19064.336 BT=1227.6031
SERIES=71.4489396 FSERIES=27710.969

I=5 DEMAND=10000.0 S=0.5540571 ATM1=19064.336
BTM1=1227.6031 AT=20067.611 BT=1182.7375
SERIES=80.5540571 FSERIES=11288.914

I=6 DEMAND=18000.0 S=0.8219918 ATM1=20067.611
BTM1=1182.7375 AT=21315.117 BT=1195.6913
SERIES=90.8219918 FSERIES=17775.572

I=7 DEMAND=23000.0 S=1.1835692 ATM1=21315.117
BTM1=1195.6913 AT=22203.002 BT=1134.13
SERIES=101.1835692 FSERIES=27074.145

I=8 DEMAND=38000.0 S=1.4299351 ATM1=22203.002
BTM1=1134.13 AT=23660.883 BT=1198.8801
SERIES=111.4299351 FSERIES=34497.184

I=9 DEMAND=12000.0 S=0.5484829 ATM1=23660.883
BTM1=1198.8801 AT=24561.64 BT=1139.2557
SERIES=120.5484829 FSERIES=13917.754

I=10 DEMAND=13000.0 S=0.8242397 ATM1=24561.64
BTM1=1139.2557 AT=24708.018 BT=940.68
SERIES=130.8242397 FSERIES=21945.38

I=11 DEMAND=32000.0 S=1.1688018 ATM1=24708.018
BTM1=940.68 AT=25821.672 BT=975.2749
SERIES=141.1688018 FSERIES=32580.35

I=12 DEMAND=41000.0 S=1.4475441 ATM1=25821.672
BTM1=975.2749 AT=26949.635 BT=1005.8125
SERIES=151.4475441 FSERIES=42159.723

FORECAST = 15165

FORECAST = 23007

FORECAST = 35236

FORECAST = 45063

BIBLIOGRAPHY

-http://en.wikipedia.org/wiki/Main_Page

-http://processing.org/

-Robust forecasting with exponential and Holt-Winters smoothing

Sarah Gelper, Roland Fried and Christophe Croux

-http://itl.nist.gov/div898/handbook/pmc/section4/pmc436.htm

-http://adorio-research.org/wordpress/?p=1230