Forecasting

in Economics, Business, Finance and Beyond

CH 9A: Forecasting Techniques

Forecasting Techniques

So far we have focused on the construction of time series models for both stationary and nonstationary data, and the calculation of minimum MSE predictors based on these models. We will discuss 3 additional forecasting techniques that have **less** emphasis on the explicit construction of a model for the data. These techniques have been found in practice to be effective on a wide range of real data sets.

We will look at the following techniques:

- ARMA
- ARAR
- Exponential Smoothing
- Holt-Winters (non-seasonal and seasonal)

2

ARMA Forecasting

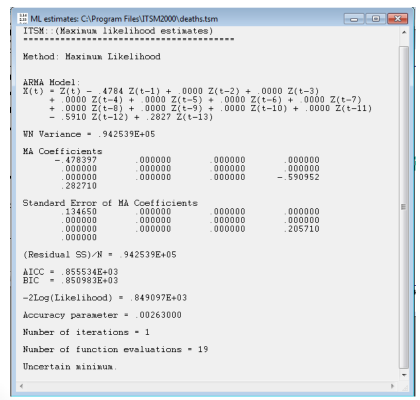
This technique just applies the modeling techniques we have been using to forecast future observations.

In ITSM simply fit your model as usual, then click **Forecasting -> ARMA**. The resulting dialog box allows you to select options so that the forecasts are in the units of the original data (these options are the defaults).

3

ARMA Forecasting

Example: DEATHS.TSM Fit SARIMA(0,1,1)x(0,1,1)$_{12}$.



```
ML estimates: C:\Program Files\ITSM2000\deaths.tsm
ITSM::(Maximum likelihood estimates)
========================================

Method: Maximum Likelihood


ARMA Model:
X(t) = Z(t) - .4784 Z(t-1) + .0000 Z(t-2) + .0000 Z(t-3)
     + .0000 Z(t-4) + .0000 Z(t-5) + .0000 Z(t-6) + .0000 Z(t-7)
     + .0000 Z(t-8) + .0000 Z(t-9) + .0000 Z(t-10) + .0000 Z(t-11)
     - .5910 Z(t-12) + .2827 Z(t-13)

WN Variance = .942539E+05

MA Coefficients
      -.478397       .000000       .000000       .000000
       .000000       .000000       .000000       .000000
       .000000       .000000       .000000      -.590952
       .282710

Standard Error of MA Coefficients
       .134650       .000000       .000000       .000000
       .000000       .000000       .000000       .000000
       .000000       .000000       .000000       .205710
       .000000

(Residual SS)/N = .942539E+05

AICC = .855534E+03
BIC  = .850983E+03

-2Log(Likelihood) = .849097E+03

Accuracy parameter = .00263000

Number of iterations = 1

Number of function evaluations = 19

Uncertain minimum.
```

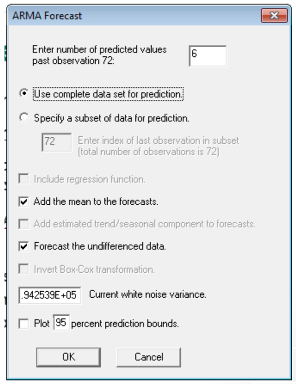

4

ARMA Forecasting

Example: DEATHS.TSM   Fit SARIMA(0,1,1)x(0,1,1)$_{12}$.

Forecasting -> ARMA

Forecast 6 values.



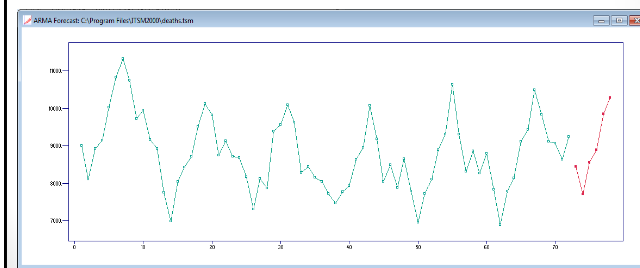
5

ARMA Forecasting

Example: DEATHS.TSM   Fit SARIMA(0,1,1)x(0,1,1)$_{12}$.

Forecasting -> ARMA

Forecast 6 values.



6

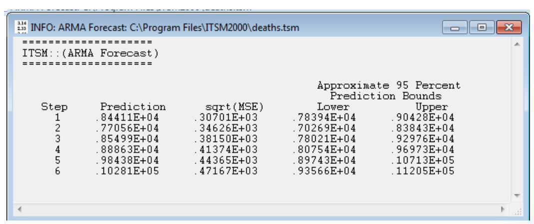
ARMA Forecasting

Example: DEATHS.TSM   Fit SARIMA(0,1,1)x(0,1,1)$_{12}$.

Forecasting -> ARMA

Forecast 6 values.

Info Window gives details



INFO: ARMA Forecast: C:\Program Files\ITSM2000\deaths.tsm

ITSM::(ARMA Forecast)

Approximate 95 Percent Prediction Bounds

Step	Prediction	sqrt(MSE)	Lower	Upper
1	.84411E+04	.30701E+03	.78394E+04	.90428E+04
2	.77056E+04	.34626E+03	.70269E+04	.83843E+04
3	.85499E+04	.38150E+03	.78021E+04	.92976E+04
4	.88863E+04	.41374E+03	.80754E+04	.96973E+04
5	.98438E+04	.44365E+03	.89743E+04	.10713E+05
6	.10281E+05	.47167E+03	.93566E+04	.11205E+05

7

ARAR Forecasting

The ARAR Algorithm has two steps:

1) Memory Shortening.

Reduces the data to a series which can reasonably be modeled as an ARMA process.

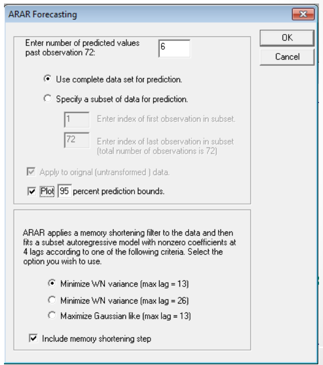
2) Fitting a Subset Autoregression.

Fits a subset AR model with lags $\{1,k_1,k_2,k_3\}$, $1<k_1<k_2<k_3\leq m$ to the memory shortened data. The lags $\{k_1,k_2,k_3\}$ and corresponding model parameters are estimated either by minimizing σ^2, or maximizing the Gaussian likelihood.

8

ARAR Forecasting

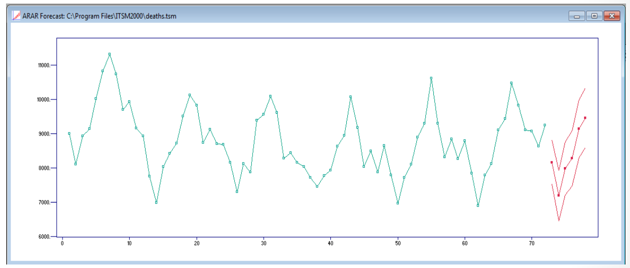
Minimum MSE forecasts can then be computed based on the fitted models.



Example: (DEATHS.TSM).
Forecasting > **ARAR**.
Forecast next 6 months using m=13 (minimize WN variance). **Info** window gives details.

ARAR Forecasting

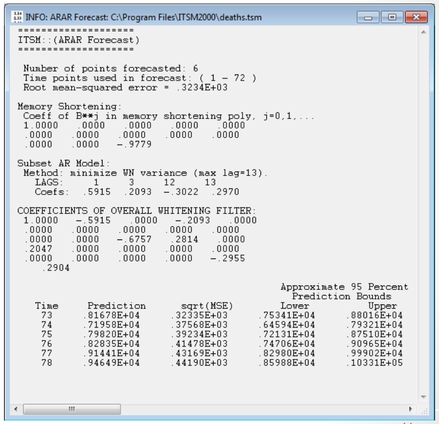
Example: (DEATHS.TSM).



ARAR Forecasting

Example:

(DEATHS.TSM).



INFO: ARAR Forecast: C:\Program Files\ITSM2000\deaths.tsm

```
====================
ITSM::(ARAR Forecast)
====================

 Number of points forecasted: 6
 Time points used in forecast: ( 1 - 72 )
 Root mean-squared error = .3234E+03

Memory Shortening:
 Coeff of B**j in memory shortening poly, j=0,1,...
 1.0000   .0000   .0000   .0000   .0000
  .0000   .0000   .0000   .0000   .0000
  .0000   .0000  -.9779

Subset AR Model:
 Method: minimize WN variance (max lag=13).
   LAGS:     1      3      12      13
   Coefs:  .5915  .2093  -.3022  .2970

COEFFICIENTS OF OVERALL WHITENING FILTER:
 1.0000  -.5915   .0000  -.2093   .0000
  .0000   .0000   .0000   .0000   .0000
  .0000   .0000  -.6757   .2814   .0000
  .2047   .0000   .0000   .0000   .0000
  .0000   .0000   .0000   .0000  -.2955
    .2904

                                           Approximate 95 Percent
                                             Prediction Bounds
   Time     Prediction      sqrt(MSE)        Lower          Upper
    73      .81678E+04     .32335E+03     .75341E+04     .88016E+04
    74      .71958E+04     .37568E+03     .64594E+04     .79321E+04
    75      .79820E+04     .39234E+03     .72131E+04     .87510E+04
    76      .82835E+04     .41478E+03     .74706E+04     .90965E+04
    77      .91441E+04     .43169E+03     .82980E+04     .99902E+04
    78      .94649E+04     .44190E+03     .85988E+04     .10331E+05
```


Exponential Smoothing

For any fixed $\alpha \in [0, 1]$, the one-sided moving averages defined by the recursions

$$\hat{m}_t = \alpha X_t + (1-\alpha)\hat{m}_{t-1}, \quad t = 2, \dots, n,$$

and
$$\hat{m}_1 = X_1$$

can be computed using ITSM by selecting **Smooth>Exponential** and specifying the value of α. Setting α = -1 instructs ITSM to select the optimum value for α.

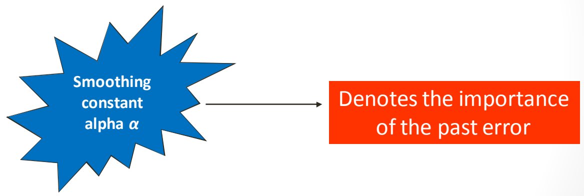
Application of these recursions is often referred to as exponential smoothing, since the recursions imply that for $t \geq 2$,

$$\hat{m}_t = \sum_{j=0}^{t-2} \alpha(1-\alpha)X_{t-j} + (1-\alpha)^{t-1} X_1$$

a weighted moving average of $X_t, X_{t-1}, \dots$, with weights decreasing exponentially (except for the last one).

Exponential Smoothing

Main idea: The prediction of the future depends mostly on the most recent observation, and on the error for the latest forecast.



13

Exponential Smoothing

Why use exponential smoothing?

1. Uses less storage space for data
2. Extremely accurate
3. Easy to understand
4. Little calculation complexity
5. There are simple accuracy tests

14

Exponential Smoothing

The smoothing constant α expresses how much our forecast will react to observed differences…

If α is low: there is little reaction to differences.

If α is high: there is a lot of reaction to differences.

Forecasting by exponential smoothing with optimal α can be viewed as fitting a member of the two-parameter family of ARIMA(0,1,1) processes given by

$$Y_t = Y_{t-1} + Z_t - (1-\alpha)Z_{t-1}, \quad \{Z_t\} \sim \mathrm{WN}\left(0, \sigma^2\right)$$

15

Exponential Smoothing

In ITSM, the optimal α is found by minimizing the average squared error of the one-step forecasts of the *observed* data $Y_2, \ldots, Y_n$, and the parameter σ^2 is estimated by this average squared error.

This algorithm could easily be modified to minimize other error measures such as

- average absolute one-step error
- average 12-step squared error.

16

Example: bottled water at Kroger

Month	Actual	Forecasted
Jan	1,325	1,370
Feb	1,353	1,361
Mar	1,305	1,359
Apr	1,275	1,349
May	1,210	1,334
Jun	?	1,309

$\alpha = 0.2$

17

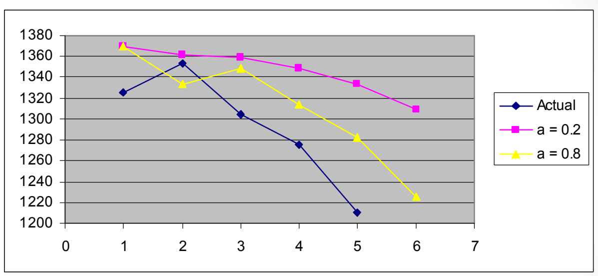
Example: bottled water at Kroger

Month	Actual	Forecasted
Jan	1,325	1,370
Feb	1,353	1,334
Mar	1,305	1,349
Apr	1,275	1,314
May	1,210	1,283
Jun	?	1,225

$\alpha = 0.8$

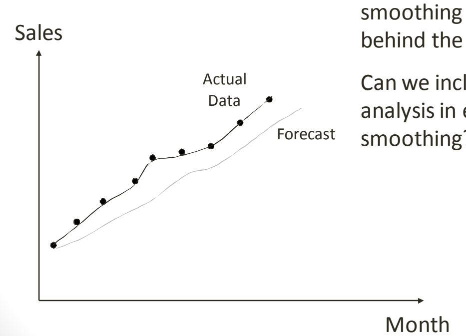
18

Impact of the smoothing constant



19

Impact of trend



Regular exponential smoothing will always lag behind the trend.

Can we include trend analysis in exponential smoothing?

20

Holt-Winters Forecasting

The Holt-Winters (HW) Algorithm

This algorithm is primarily suited for series that have a locally linear trend but no seasonality. The basic idea is to allow for a time-varying trend by specifying the forecasts to have the form:

$$P_t Y_{t+h} = \hat{a}_t + \hat{b}_t h, \quad h = 1,2,3,\ldots$$

where $\hat{a}_t$ is the estimated **level** at time t, and

$\hat{b}_t$ is the estimated **slope** at time t.

The Holt-Winters (HW) Algorithm

Like *exponential smoothing,* we now take the estimated level at time *t*+1 to be a weighted average of the observed and forecast values, i.e.

$$\hat{a}_{t+1} = \alpha\, Y_{t+1} + (1-\alpha) P_t Y_{t+1} = \alpha\, Y_{t+1} + (1-\alpha)(\hat{a}_t + \hat{b}_t).$$

Similarly, the estimated slope at time t+1 is given by

$$\hat{b}_{t+1} = \beta(\hat{a}_{t+1} - \hat{a}_t) + (1-\beta)\hat{b}_t.$$

The Holt-Winters (HW) Algorithm

With the natural initial conditions,

$\hat{a}_2 = Y_2,$ and $\hat{b}_2 = Y_2 - Y_1,$

and by choosing α and β to minimize the sum of squares of the one-step prediction errors,

$$(Y_t - P_{t-1}Y_t)^2,$$

the recursions for $\hat{a}_2$ and $\hat{b}_2$ can be solved for *t* = 2,…,n.

The forecasts then have the form:

$$P_n Y_{n+h} = \hat{a}_n + \hat{b}_n h, \quad h = 1,2,3,\ldots.$$

The Holt-Winters (HW) Algorithm

Forecasting by exponential smoothing with optimal α can be viewed as fitting a member of the two-parameter family of ARIMA(0,1,1) processes given by

$$Y_t = Y_{t-1} + Z_t - (1-\alpha)Z_{t-1}, \quad \{Z_t\} \sim \mathrm{WN}\left(0, \sigma^2\right)$$

Holt–Winters forecasting can be viewed as fitting a member of the three-parameter family of ARIMA processes

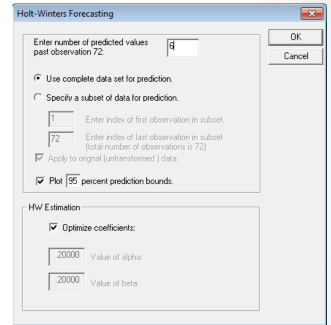
$$(1-B)^2 Y_t = Z_t - (2-\alpha-\alpha\beta)Z_{t-1} + (1-\alpha)Z_{t-2}$$

$$\{Z_t\} \sim \mathrm{WN}\left(0, \sigma^2\right)$$

25

The Holt-Winters (HW) Algorithm

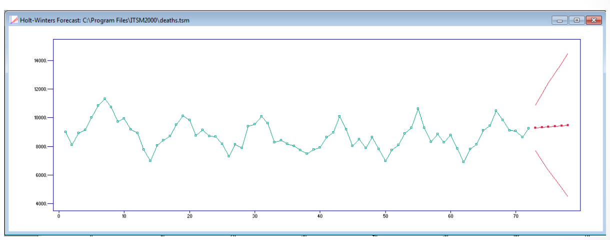
Example: DEATHS.TSM
Forecasting > Holt-Winters.
Forecast next 6 months.
Info window gives details.



26

The Holt-Winters (HW) Algorithm

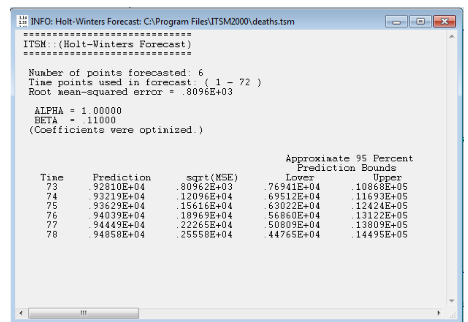
Example: DEATHS.TSM
Forecasting > Holt-Winters. Forecast next 6 months.
Note that this method ignores seasonality.



27

The Holt-Winters (HW) Algorithm

Example: DEATHS.TSM
Forecasting > Holt-Winters. Forecast next 6 months.
Info window gives details.



INFO: Holt-Winters Forecast: C:\Program Files\ITSM2000\deaths.tsm

```
==============================
ITSM::(Holt-Winters Forecast)
==============================

 Number of points forecasted: 6
 Time points used in forecast: ( 1 - 72 )
 Root mean-squared error = .8096E+03

  ALPHA = 1.00000
  BETA  = .11000
 (Coefficients were optimized.)


                                         Approximate 95 Percent
                                           Prediction Bounds
   Time     Prediction      sqrt(MSE)      Lower          Upper
    73      .92810E+04     .80962E+03    .76941E+04     .10868E+05
    74      .93219E+04     .12096E+04    .69512E+04     .11693E+05
    75      .93629E+04     .15616E+04    .63022E+04     .12424E+05
    76      .94039E+04     .18969E+04    .56860E+04     .13122E+05
    77      .94449E+04     .22265E+04    .50809E+04     .13809E+05
    78      .94858E+04     .25558E+04    .44765E+04     .14495E+05
```

28

The Seasonal Holt-Winters (HW) Algorithm

It's clear from the previous example that the HW Algorithm does not handle series with seasonality very well. If we know the period (d) of our series, HW can be modified to take this into account. In this seasonal version of HW, the forecast function is modified to:

$$P_t Y_{t+h} = \hat{a}_t + \hat{b}_t h + \hat{c}_{t+h}, \quad h = 1,2,3,\ldots$$

where $\hat{a}_t$ and $\hat{b}_t$ are as before, and $\hat{c}_t$ is the estimated *seasonal component* at time t.

29

The Seasonal Holt-Winters (HW) Algorithm

With the same recursions for $\hat{b}_t$ as in HW, we modify the recursion for $\hat{a}_t$ according to

$$\hat{a}_{t+1} = \alpha(Y_{t+1} - \hat{c}_{t+1-d}) + (1-\alpha)(\hat{a}_t + \hat{b}_t),$$

and add the additional recursion for $\hat{c}_t$

$$\hat{c}_{t+1} = \gamma(Y_{t+1} - \hat{a}_{t+1}) + (1-\gamma)\hat{c}_{t+1-d}.$$

30

The Seasonal Holt-Winters (HW) Algorithm

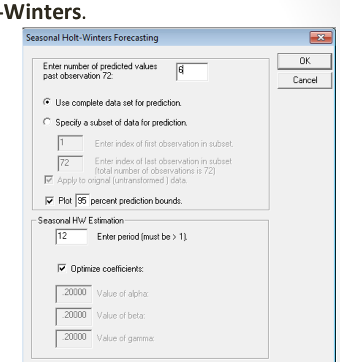
Analogous to HW, natural initial conditions hold to start off the recursions, and the smoothing parameters $\{\alpha, \beta, \gamma\}$, are once again chosen to minimize the sum of squares of the one-step prediction errors. The forecasts then have the form:

$$P_n Y_{n+h} = \hat{a}_n + \hat{b}_n h + \hat{c}_{n+h}, \quad h = 1,2,3,\ldots$$

31

The Seasonal Holt-Winters (HW) Algorithm

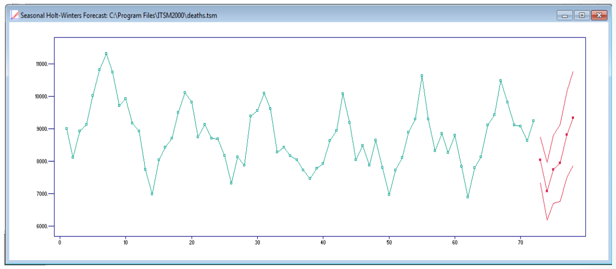
Example: (DEATHS.TSM).
Forecasting > **Seasonal Holt-Winters**.
Forecast next 6 months.
Info window gives details.



32

The Seasonal Holt-Winters (HW) Algorithm

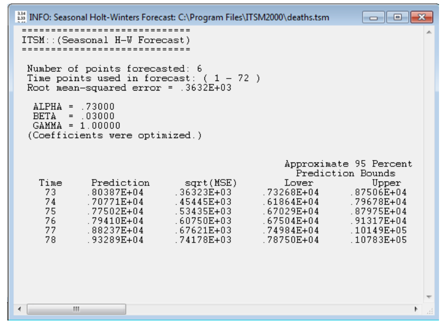
Example: (DEATHS.TSM). **Forecasting** > **Seasonal Holt-Winters**.
Forecast next 6 months.



33

The Seasonal Holt-Winters (HW) Algorithm

Example: (DEATHS.TSM).
Forecasting > **Seasonal Holt-Winters**.
Forecast next 6 months. **Info** window gives details.



INFO: Seasonal Holt-Winters Forecast: C:\Program Files\ITSM2000\deaths.tsm

```
================================
ITSM::(Seasonal H-W Forecast)
================================

 Number of points forecasted: 6
 Time points used in forecast: ( 1 - 72 )
 Root mean-squared error = .3632E+03

  ALPHA = .73000
  BETA  = .03000
  GAMMA = 1.00000
 (Coefficients were optimized.)


                                              Approximate 95 Percent
                                               Prediction Bounds
   Time      Prediction        sqrt(MSE)         Lower          Upper
    73       .80387E+04       .36323E+03      .73268E+04     .87506E+04
    74       .70771E+04       .45445E+03      .61864E+04     .79678E+04
    75       .77502E+04       .53435E+03      .67029E+04     .87975E+04
    76       .79410E+04       .60750E+03      .67504E+04     .91317E+04
    77       .88237E+04       .67621E+03      .74984E+04     .10149E+05
    78       .93289E+04       .74178E+03      .78750E+04     .10783E+05
```

34