

Homework 7 Solutions

Due: Thursday 4/18/19 by 12:00pm (noon)

Note - problems 2. and 3. will require use of the the MARSS package for R.

Grading Scheme:

- Maximum of 2 points for 1., determined as follows:
 - 0 points for no solutions or R output only;
 - 1 point if there are substantial mistakes and serious revisions needed for final project draft;
 - 2 points if mostly or entirely correct and only minor revisions needed for final project draft.
- 1 point for correct or nearly correct answers to 2.a., and for 3.a.-c. (combined).
- Maximum of 3 points each for 2.a. and 3.b, determined as follows:
 - 0 points for no solutions whatsoever or R output only;
 - 1 point for an honest effort but very few correct answers;
 - 2 points for mostly correct answers but some major mistakes;
 - 3 points for one or fewer mistakes.

Your total score will be divided by 2, to make this homework weighted equally with the rest of the homeworks. Solutions for problems 2. and 3. are given below in blue, since you will be asked to improve upon your solution to 1. for your final project.

2. Understanding State Space Models

In this problem, we're going to play around with the parameters of the state-space model a bit, building off of the example of the `varve` data we saw in class. The `varve` data is available in the `astsa` package. We'll work with it on the log scale, because the variance of the observations appears to be increasing over time. In class, we fit the following model to the data:

$$y_t = ax_t + v_t \text{ and } x_t = \phi x_{t-1} + w_t,$$

where $v_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma_v^2)$, $w_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2)$, and $x_0 = \mu$.

```
library(astsa)
library(MARSS)

data("varve")
n <- length(varve)

y <- log(varve)

model <- list(
  B=matrix("phi"), U=matrix(0), Q=matrix("sig.sq.w"),
  Z=matrix("a"), A=matrix(0), R=matrix("sig.sq.v"),
  x0=matrix("mu"), tinitx=0 )

fit <- MARSS(c(y), model=model, method = "kem", silent=TRUE)
# Get Kalman filter, predictor and smoother
kf <- MARSSkfss(fit)
```

The estimated coefficients were $\hat{a} = 0.9776$, $\hat{\sigma}_v^2 = 0.1805$, $\hat{\phi} = 0.999$, $\hat{\sigma}_w^2 = 0.0133$, and $\hat{\mu} = 3.5267$.

For this problem, I would like you to plot the predictions, filter, and smoother for the following state-space models:

- i. Keeping \hat{a} , $\hat{\phi}$, $\hat{\sigma}_w^2$, and $\hat{\mu}$ at their estimated values and setting $\sigma_v^2 = 0.05$.
 - ii. Keeping \hat{a} , $\hat{\phi}$, $\hat{\sigma}_w^2$, and $\hat{\mu}$ at their estimated values and setting $\sigma_v^2 = 0.5$.
 - iii. Keeping \hat{a} , $\hat{\phi}$, $\hat{\sigma}_v^2$, and $\hat{\mu}$ at their estimated values and setting $\sigma_w^2 = 0.0001$.
 - iv. Keeping \hat{a} , $\hat{\phi}$, $\hat{\sigma}_v^2$, and $\hat{\mu}$ at their estimated values and setting $\sigma_w^2 = 1$.
- (a) Make a plot with twelve panels using `par(mfrow = c(4, 3))`. In each panel, plot the first 50 observations from the data in gray.
- In the first column of panels, add solid black lines for the predictions of the states using the estimated parameters and dashed black lines for the predictions of the states using the estimated parameters plus and minus the corresponding standard errors.
 - In the second column of panels, add solid black lines for the filtered values of the states using the estimated parameters and dashed black lines for the filtered values of the states using the estimated parameters plus and minus the corresponding standard errors.
 - In the third column of panels, add solid black lines for the smoothed values of the states using the estimated parameters and dashed black lines for the smoothed values of the states using the estimated parameters plus and minus the corresponding standard errors.
 - In the first row of panels, add lines for the the predictions, filtered values, and smoothed values from model i.
 - In the second row of panels, add lines for the the predictions, filtered values, and smoothed values from model ii.
 - In the third row of panels, add lines for the the predictions, filtered values, and smoothed values from model iii.
 - In the fourth row of panels, add lines for the the predictions, filtered values, and smoothed values from model iv.

```
models <- list(
  list(B=matrix(coef(fit)$B[1, 1]), U=matrix(0), Q=matrix(coef(fit)$Q[1, 1]),
    Z=matrix(coef(fit)$Z[1, 1]), A=matrix(0), R=matrix(0.05),
    x0=matrix(coef(fit)$x0[1, 1]), tinitx=0 ),
  list(B=matrix(coef(fit)$B[1, 1]), U=matrix(0), Q=matrix(coef(fit)$Q[1, 1]),
    Z=matrix(coef(fit)$Z[1, 1]), A=matrix(0), R=matrix(0.5),
    x0=matrix(coef(fit)$x0[1, 1]), tinitx=0 ),
  list(B=matrix(coef(fit)$B[1, 1]), U=matrix(0), Q=matrix(0.0001),
    Z=matrix(coef(fit)$Z[1, 1]), A=matrix(0), R=matrix(coef(fit)$R[1, 1]),
    x0=matrix(coef(fit)$x0[1, 1]), tinitx=0 ),
  list(B=matrix(coef(fit)$B[1, 1]), U=matrix(0), Q=matrix(1),
    Z=matrix(coef(fit)$Z[1, 1]), A=matrix(0), R=matrix(coef(fit)$R[1, 1]),
    x0=matrix(coef(fit)$x0[1, 1]), tinitx=0 ))

sub <- 1:50

par(mfrow = c(4, 3))
par(mar = rep(2, 4))
for (mod in models) {

  fit.mod <- MARSS(c(y), model=mod, method = "kem", silent=TRUE)
  # Get Kalman filter, predictor and smoother
  kf.mod <- MARSSkfss(fit.mod)

  plot(c(y)[sub],
    main = "Kalman Predictions", xlab = "Time", ylab = "y",
    col = "darkgray", type = "b", pch = 16, lty = 1)
```

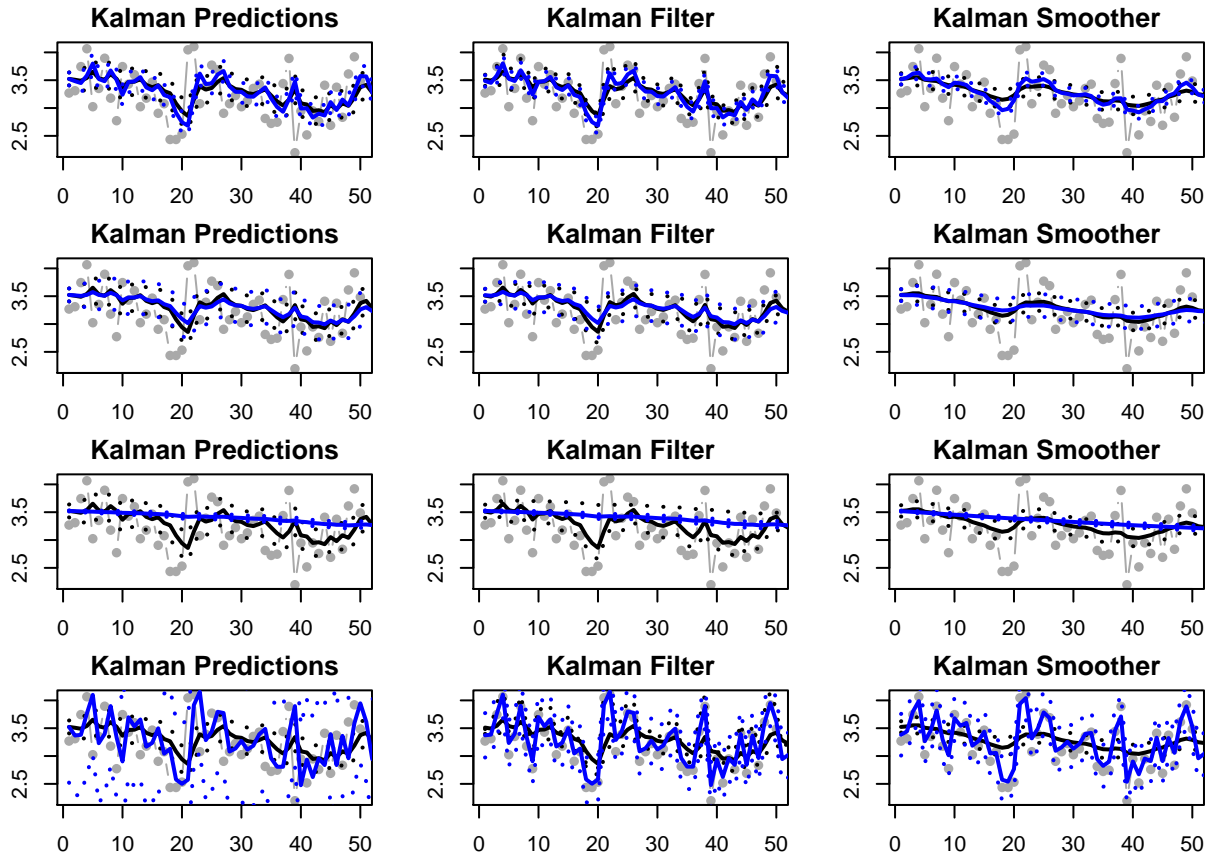
```

lines(c(kf[["xtt1"]]), col = "black", lwd = 2)
lines(c(kf[["xtt1"]]) - c(sqrt(kf[["Vtt1"]])), col = "black", lwd = 2, lty = 3)
lines(c(kf[["xtt1"]]) + c(sqrt(kf[["Vtt1"]])), col = "black", lwd = 2, lty = 3)
lines(c(kf.mod[["xtt1"]]), col = "blue", lwd = 2)
lines(c(kf.mod[["xtt1"]]) - c(sqrt(kf.mod[["Vtt1"]])), col = "blue", lwd = 2, lty = 3)
lines(c(kf.mod[["xtt1"]]) + c(sqrt(kf.mod[["Vtt1"]])), col = "blue", lwd = 2, lty = 3)

plot(c(y)[sub],
      main = "Kalman Filter", xlab = "Time", ylab = "y",
      col = "darkgray", type = "b", pch = 16, lty = 1)
lines(c(kf[["xtt"]]), col = "black", lwd = 2)
lines(c(kf[["xtt"]]) - c(sqrt(kf[["Vtt"]])), col = "black", lwd = 2, lty = 3)
lines(c(kf.mod[["xtt"]]) + c(sqrt(kf[["Vtt"]])), col = "black", lwd = 2, lty = 3)
lines(c(kf.mod[["xtt"]]), col = "blue", lwd = 2)
lines(c(kf.mod[["xtt"]]) - c(sqrt(kf.mod[["Vtt"]])), col = "blue", lwd = 2, lty = 3)
lines(c(kf.mod[["xtt"]]) + c(sqrt(kf.mod[["Vtt"]])), col = "blue", lwd = 2, lty = 3)

plot(c(y)[sub], main = "Kalman Smoother", xlab = "Time", ylab = "y",
      col = "darkgray", type = "b", pch = 16, lty = 1)
lines(c(fit$states), col = "black", lwd = 2)
lines(c(fit$states) - c(fit$states.se), col = "black", lwd = 2, lty = 3)
lines(c(fit$states) + c(fit$states.se), col = "black", lwd = 2, lty = 3)
lines(c(fit.mod$states), col = "blue", lwd = 2)
lines(c(fit.mod$states) - c(fit.mod$states.se), col = "blue", lwd = 2, lty = 3)
lines(c(fit.mod$states) + c(fit.mod$states.se), col = "blue", lwd = 2, lty = 3)
}

```



- (b) In at one sentence, explain how the value of σ_v^2 affects the conditional means and variances of the states given the observed data.

Decreasing σ_v^2 makes the conditional means of the states more closely track the observed data, and it makes the conditional variances of the states smaller.

- (c) In at one sentence, explain how the value of σ_w^2 affects the conditional means and variances of the states given the observed data.

Decreasing σ_w^2 makes the conditional means of the states vary more smoothly and less like the observed data, and makes the conditional variances of the states smaller.

3. A State Space Model for the Chicken Data

In this problem, we're going to further examine the use of the state-space model

$$y_t = ax_t + v_t \text{ and } x_t = \phi x_{t-1} + w_t,$$

where $v_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma_v^2)$, $w_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2)$, and $x_0 = \mu$.

We will use the `chicken` data from the `astsa` package once more. A special case of this state-space model is the **ARIMA**(1,0,0) model.

- (a) Fit the state space model to the `chicken` data with $a = 1$ and $\sigma_v^2 = 0$, as well as an **ARIMA**(1,0,0) model. You will need to use `method="ML"` when you run `arima` to fit this model. Compare the parameter estimates. In at most one sentence, indicate whether or not the estimates of the remaining parameters are the same and explain why or why not this is the case.

```

data("chicken")
n <- length(chicken)

y <- c(chicken[1:(n - 20)], rep(NA, 20))

model <- list(B=matrix("phi"), U=matrix(0), Q=matrix("sig.sq.w"),
              Z=matrix(1), A=matrix(0), R=matrix(0),
              x0=matrix("mu"), tinitx=0 )

fit <- MARSS(c(y), model=model, method = "kem", silent=TRUE)

ar <- arima(c(na.omit(y)), order = c(1, 0, 0), method = "ML")

```

The estimates are not the same because these models are similar but not quite the same - no fixed initial state is assumed by the ARIMA model.

- (b) Fit the entire state space model with unknown a and σ_v^2 to the chicken data with the last 20 observations removed. On a plot with the x -axis ranging from $2001 + 7/12$ to $2016 + 6/12$ and y -axis ranging from 60 to 130, plot:

- The observed chicken time series.
- Predicted chicken prices from $2014 + 11/12$ to $2016 + 6/12$ based on the **ARIMA**(1,0,0) model.
- Predicted chicken prices from $2014 + 11/12$ to $2016 + 6/12$ based on the **ARIMA**(1,0,0) model plus or minus one standard error.
- Forecasted chicken prices from $2014 + 11/12$ to $2016 + 6/12$ based on the state-space model.
- Forecasted chicken prices from $2014 + 11/12$ to $2016 + 6/12$ based on the state-space model plus or minus one standard error.

```

model <- list(B=matrix("phi"), U=matrix(0), Q=matrix("sig.sq.w"),
              Z=matrix("a"), A=matrix(0), R=matrix("sig.sq.v"),
              x0=matrix("mu"), tinitx=0 )

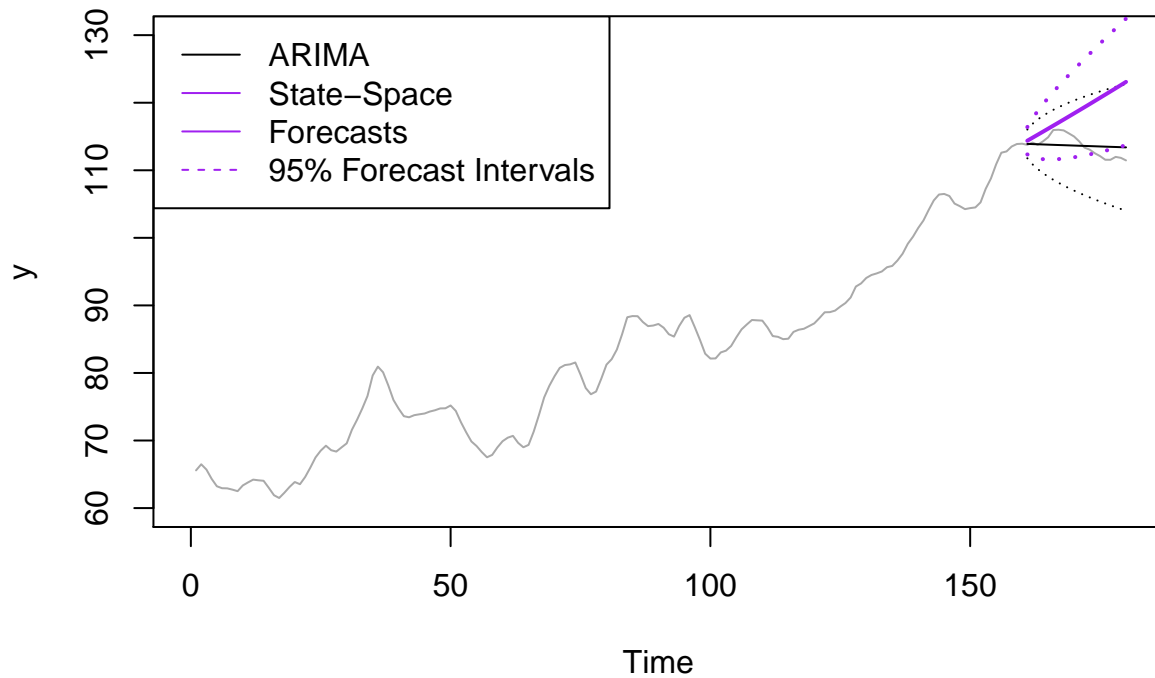
fit <- MARSS(c(y), model=model, method = "kem", silent = TRUE)
fit.bfgs <- MARSS(c(y), model=model, method = "BFGS",
                  inits = fit, silent = TRUE)
# Get Kalman filter, predictor and smoother
kf <- MARSSkfss(fit.bfgs)

par(mfrow = c(1, 1))
plot(c(chicken), type = "l", col = "darkgray", main = "Kalman Smoother",
     xlab = "Time", ylab = "y",
     xlim = c(0, length(y)),
     ylim = c(60, 130))
pred <- predict(ar, n.ahead = 20)
lines((n - 20 + 1):n, pred$pred)
lines((n - 20 + 1):n, pred$pred - qnorm(0.975)*pred$se,
     lty = 3)
lines((n - 20 + 1):n, pred$pred + qnorm(0.975)*pred$se,
     lty = 3)
lines((n - 20 + 1):n, (coef(fit)$Z[1, 1]*c(fit.bfgs$states))[(n - 20 + 1):n],
     col = "purple", lwd = 2)
lines((n - 20 + 1):n, ((coef(fit)$Z[1, 1]*fit.bfgs$states) -
                      qnorm(0.975)*c(sqrt(coef(fit)$Z[1, 1]^2*fit.bfgs$states.se^2 +
                      coef(fit)$R[1, 1]))))[(n - 20 + 1):n],
     col = "purple", lwd = 2, lty = 3)

```

```
lines((n - 20 + 1):n, ((c(coef(fit)$Z[1, 1]*c(fit.bfgs$states) +
                          qnorm(0.975)*c(sqrt(coef(fit)$Z[1, 1]^2*fit.bfgs$states.se^2 +
                          coef(fit)$R[1, 1]))))[(n - 20 + 1):n],
      col = "purple", lwd = 2, lty = 3)
legend("topleft", col = c("black", "purple", "purple", "purple"),
      lty = c(1, 1, 1, 2),
      legend = c("ARIMA", "State-Space", "Forecasts", "95% Forecast Intervals"))
```

Kalman Smoother



(c) In at most one sentence, indicate which model you prefer for the chicken data and explain why.

Any answer accompanied by sound reasoning would be accepted here. For instance:

- I would prefer the state-space model, because the forecasts reflect the overall trend of the time series and the 95% intervals do contain most of the true values.
- I would prefer the ARIMA model, because the forecasts are more similar to the observed values.