Lab 5.3: Spatio-Temporal Inference with Unknown Evolution Operator

If we have no prior knowledge to guide us on how to parameterize $\mathbf{M}$, then $\mathbf{M}$ can be estimated in full in the context of a standard state-space modeling framework. When taking this approach, it is important that a very low-dimensional representation of the spatio-temporal process is adopted – the dimension of the parameter space increases quadratically with the dimension of the process, and thus the model can easily become over-parameterized.

Empirical orthogonal functions (EOFs) are ideal basis functions to use in this case, since they capture most of the variability in the observed signal, by design. In this Lab we look at the SST data set, take the EOFs that we generated in Lab 2.3, and estimate all unknown parameters, first within a classical time-series framework based on a vector autoregression and using the method of moments ,and then in a state-space framework using the EM algorithm.

Time-Series Framework

The aim of this first part of the Lab is to show how even simple methods can be used in a dynamical setting to provide prediction and prediction standard errors on a variable of interest. These methods work particularly well when we have complete spatial coverage and a high signal-to-noise ratio; this is the case with the SST data.

First, we load the usual packages.

```r
library("ggplot2")
library("STRbook")
```

Then we load **expm** for raising matrices to a specified power and **Matrix**, which here we only use for plotting purposes.

```r
library("expm")
library("Matrix")
```

We now load the SST data, but this time we truncate it at April 1997 in order to forecast the SSTs 6 months ahead, in October 1997.

```r
data("SSTlandmask", package = "STRbook")
data("SSTlonlat", package = "STRbook")
data("SSTdata", package = "STRbook")
delete_rows <- which(SSTlandmask == 1)    # remove land values
SST_Oct97 <- SSTdata[-delete_rows, 334]    # save Oct 1997 SSTs
SSTdata <- SSTdata[-delete_rows, 1:328]    # until April 1997
SSTlonlat$mask <- SSTlandmask              # assign mask to df
```

Next, we construct the EOFs using only data up to April 1997. The following code follows closely what was done in Lab 2.3, where the entire data set was used.

```
Z <- t(SSTdata)                          # data matrix
spat_mean <- apply(SSTdata, 1, mean)     # spatial mean
nT <- ncol(SSTdata)                      # no. of time points
Zspat_detrend <- Z - outer(rep(1, nT),   # detrend data
                           spat_mean)
Zt <- 1/sqrt(nT-1)*Zspat_detrend         # normalize
E <- svd(Zt)                             # SVD
```

The number of EOFs we use here to model the SST data is $n = 10$. These 10 leading EOFs capture 74% of the variability in the data.

```
n <- 10
```

Recall that the object E contains the SVD, that is, the matrices $\mathbf{U}$ and $\mathbf{V}$ and the singular values. The dimension-reduced time series of coefficients are given by the EOFs multiplied by the spatially detrended data, that is, $\boldsymbol{\alpha}_t = \boldsymbol{\Phi}'(\mathbf{Z}_t - \hat{\boldsymbol{\mu}})$, $t = 1, \ldots, T$, where $\hat{\boldsymbol{\mu}} = (1/T) \sum_{t=1}^{T} \mathbf{Z}_t$ is the estimated spatial mean.

```
TS <- Zspat_detrend %*% E$v[, 1:n]
summary(colMeans(TS))

##       Min.    1st Qu.    Median      Mean    3rd Qu.       Max.
## -1.48e-16  -5.14e-17  3.60e-18  2.30e-17  1.15e-16  2.39e-16
```

In the last line above, we have verified that the time series have mean zero, which is needed to compute covariances by taking outer products. Next, we estimate the matrices $\mathbf{M}$ and $\mathbf{C}_\eta$ using the method of moments. First we create two sets of time series that are shifted by τ time points with respect to each other; in this case we let $\tau = 6$, so that we analyze dynamics on a six-month scale. The ith column in TStplustau below corresponds to the time series at the $(i + 6)$th time point, while that in TSt corresponds to the time series at ith time point.

```
tau <- 6
nT <- nrow(TS)
TStplustau <- TS[-(1:tau), ] # TS with first tau time pts removed
TSt <- TS[-((nT-5):nT), ]    # TS with last tau time pts removed
```

The lag-0 empirical covariance matrix and the lag-τ empirical cross-covariance matrices are now computed by taking their matrix cross-product and dividing by the appropriate number of time points.

```
Cov0 <- crossprod(TS)/nT
Covtau <- crossprod(TStplustau,TSt)/(nT - tau)
```

The estimates for $\mathbf{M}$ and $\mathbf{C}_\eta$ can now be estimated from these empirical covariance matrices.

```
C0inv <- solve(Cov0)
Mest <- Covtau %*% C0inv
Ceta <- Cov0 - Covtau %*% C0inv %*% t(Covtau)
```

There are more efficient ways to compute the quantities above that ensure symmetry and positive-definiteness of the results. In particular, the inverse rarely needs to be found explicitly. For further information, the interested reader is referred to standard books on linear algebra (see, for example, Schott, 2017).

The matrices can be visualized using the function **image**.

```
image(Mest)
image(Ceta)
```

From visual inspection, the estimate of the propagator matrix, Mest, is by no means diagonally dominant, implying that there is benefit in assuming interactions between the EOFs across time steps.Further, the estimated variances along the diagonal of the covariance matrix of the additive disturbance in the IDE model, $\mathbf{C}_\eta$, decrease with the EOF index; this is expected as EOFs with higher indices tend to have higher-frequency components.

Forecasting using this EOF-reduced model is straightforward as we take the coefficients at the final time point, $\boldsymbol{\alpha}_t$, propagate those forward, and re-project onto the original space. For example, $\hat{\boldsymbol{\mu}} + \boldsymbol{\Phi}\mathbf{M}^2\boldsymbol{\alpha}_t$ gives a one-year forecast. Matrix powers (which represent multiple matrix multiplications and do not come from elementwise multiplications) can be implemented using the operator `%^%` from the package **expm**; this will be used when we implement the state-space model below. Here we consider six-month-ahead forecasts; in the code below we project ahead the EOF coefficients of the time series at the 328th time point (which corresponds to April 1997) six months into the future.

```
SSTlonlat$pred <- NA
alpha_forecast <- Mest %*% TS[328, ]
```

The projection onto the original space is done by pre-multiplying by the EOFs and adding back the estimated spatial mean.

```
idx <- which(SSTlonlat$mask == 0)
SSTlonlat$curr[idx]  <- as.numeric(E$v[, 1:n] %*% TS[328, ] +
                                     spat_mean)
SSTlonlat$pred[idx]  <- as.numeric(E$v[, 1:n] %*% alpha_forecast +
                                     spat_mean)
```

Now we add the data to the data frame for plotting purposes.

```
SSTlonlat$obs1[idx]   <- SSTdata[, 328]
SSTlonlat$obs2[idx]   <- SST_Oct97
```

The six-month-ahead prediction variances can also be computed.

```
C <- Mest %*% Cov0 %*% t(Mest) + Ceta
```

The prediction variances are found by projecting the covariance matrix `C` onto the original space and extracting the diagonal elements. The prediction standard errors are the square root of the prediction variances and hence obtained as follows.

```
SSTlonlat$predse[idx] <-
    sqrt(diag(E$v[, 1:n] %*% C %*% t(E$v[, 1:n])))
```

Plotting proceeds in a straightforward fashion using **ggplot2**. In Figure 1 we show the April 1997 data and the EOF projection for that month, as well as the October 1997 data and the forecast for that month. From visual inspection, the El Niño pattern of high SSTs is captured but the predicted anomaly is too low. This result is qualitatively similar to what we obtained in Lab 3.3 using linear regression models.

State-Space Framework

The function **DSTM_EM**, provided with the package **STRbook**, runs the EM algorithm that carries out maximum likelihood estimation in a state-space model. The function takes the data `Z`, the initial covariance $\mathbf{C}_0$ in `Cov0`, the initial state $\boldsymbol{\mu}_0$ in `muinit`, the evolution operator $\mathbf{M}$ in `M`, the covariance matrix $\mathbf{C}_\eta$ in `Ceta`, the measurement-error variance σ_ϵ^2 in `sigma2_eps`, the matrix $\mathbf{H}$ in `H`, the maximum number of EM iterations in `itermax`, and the tolerance in `tol` (the tolerance is the smallest change in the log-likelihood, multiplied by 2, required across two consecutive iterations of the EM algorithm, before terminating). All parameters supplied to the function need to be initial guesses (usually those from the method of moments suffice); these will be updated using the EM algorithm.

```
DSTM_Results <- DSTM_EM(Z = SSTdata,
                        Cov0 = Cov0,
                        muinit = matrix(0, n, 1),
                        M = Mest,
                        Ceta = Ceta,
                        sigma2_eps = 0.1,
                        H = H <- E$v[, 1:n],
                        itermax = 10,
                        tol = 1)
```
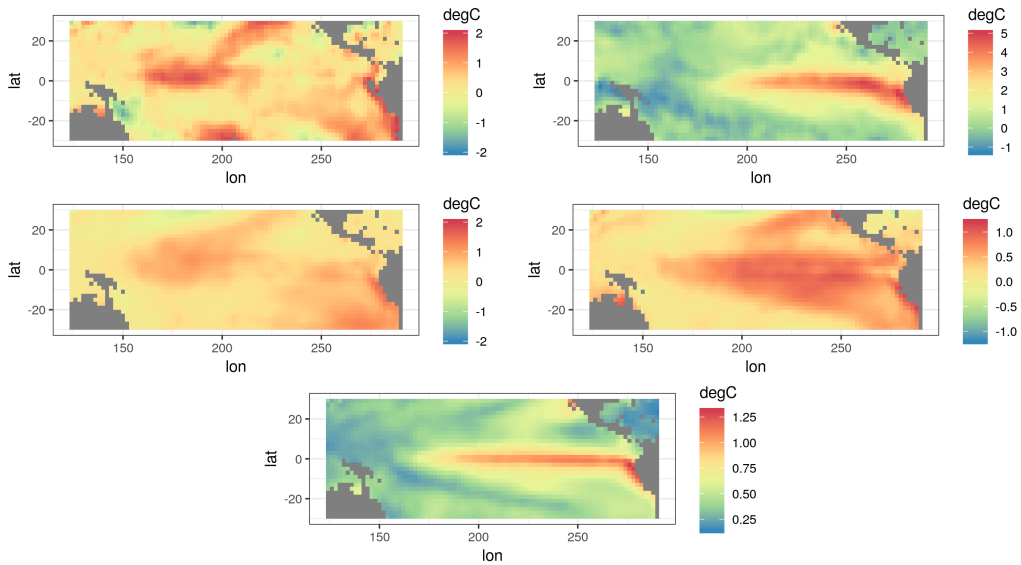
Figure 1: Top: SST data for April 1997 (left) and October 1997 (right). Middle: the EOF projection for April 1997 (left), and the forecast for October 1997 (right). Note the different color scales for the predictions (up to $1°C$) and for the observations (up to $5°C$). Bottom: Prediction standard errors for the forecast.

```
## Warning in readChar(con, 5L, useBytes = TRUE): cannot open compressed
file 'STRbook/data/DSTM_EM_results.rda', probable reason 'No such file
or directory'
## Error in readChar(con, 5L, useBytes = TRUE): cannot open the connection
```

The returned object `DSTM_Results` contains the estimated parameters, the smoothed states and their covariances, and the complete-data negative log-likelihood. In this case, estimates of $\{\alpha_t\}$ using the state-space framework are practically identical to those obtained using the time-series framework presented in the first part of the Lab. We plot estimates of $\alpha_{1,t}$, $\alpha_{2,t}$, and $\alpha_{3,t}$ below for the two methods; see Figure 2.

```
par(mfrow = c(1,3))
for(i in 1:3) {
  plot(DSTM_Results$alpha_smooth[i, ], type = 'l',
       xlab = "t", ylab = bquote(alpha[.(i)]))
  lines(TS[, i], lty = 'dashed', col = 'red')
}
```

Let us turn now to inference on the parameters. Note that the EM algorithm utilizes a Kalman filter that processes the data one time period (e.g., month) at a time. (Recall that
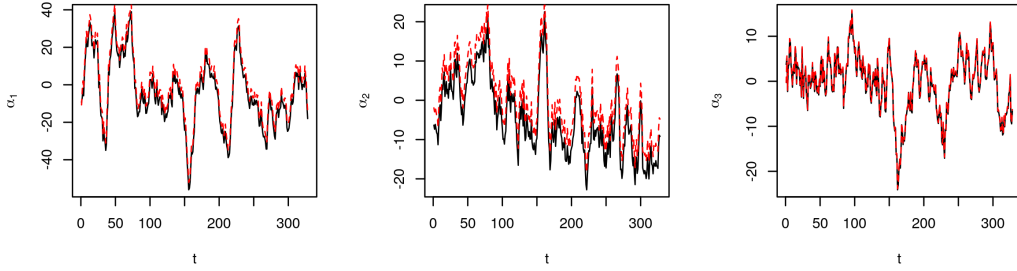
Figure 2: Estimates of $\alpha_{1,t}$ (left), $\alpha_{2,t}$ (center), and $\alpha_{3,t}$ (right) using the method of moments (red dashed line) and the EM algorithm (black solid line).

with the method of moments we let $\tau = 6$ months, so we estimated directly the state transitions over six months.) Therefore, inferences on the parameters and their interpretations differ considerably. For example, the left and right panels of Figure 3 show the estimates of the evolution matrix for the two methods. At first sight, it appears that the matrix estimated using the EM algorithm is indicating a random-walk behavior. However, if we multiply the matrix $\mathbf{M}$ by itself six times (which then describes the evolution over six months), we obtain something that is relatively similar to what was estimated using the method of moments using a time lag of $\tau = 6$ months.

To make the plots in Figure 3, we first cast the matrices into objects of class `Matrix`. Note that using the function **`image`** on objects of class `matrix` generates similar plots that are, however, less informative. On the other hand, plots of **Matrix** objects are done using the function **`levelplot`** in the **lattice** package.

```
image(as(DSTM_Results$Mest, "Matrix"))
image(as(DSTM_Results$Mest %^% 6, "Matrix"))
image(as(Mest, "Matrix"))
```

Forecasting proceeds the same way as in the method of moments. Specifically, we take the last smoothed time point (which corresponds to April 1997) and use the EM-estimated one-month propagator matrix to forecast the SSTs six months ahead. This is implemented easily using a `for` loop.

```
alpha <- DSTM_Results$alpha_smooth[, nT]
P <- DSTM_Results$Cov0
for(t in 1:6) {
   alpha <- DSTM_Results$Mest %*% alpha
   P <- DSTM_Results$Mest %*% P %*% t(DSTM_Results$Mest) +
```
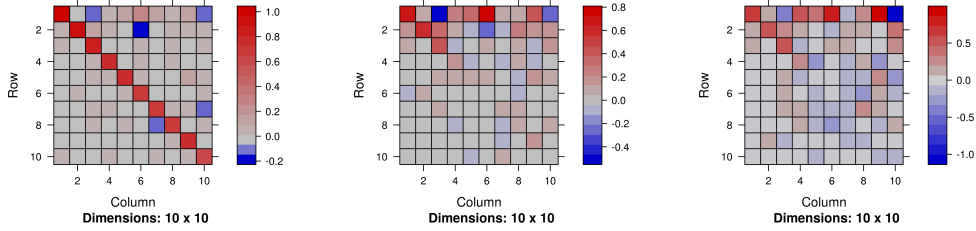
Figure 3: Estimate of a one-step-ahead evolution operator using the EM algorithm (left); EM estimate raised to the sixth power (center); and estimate of the six-steps-ahead evolution operator using the method of moments (right).
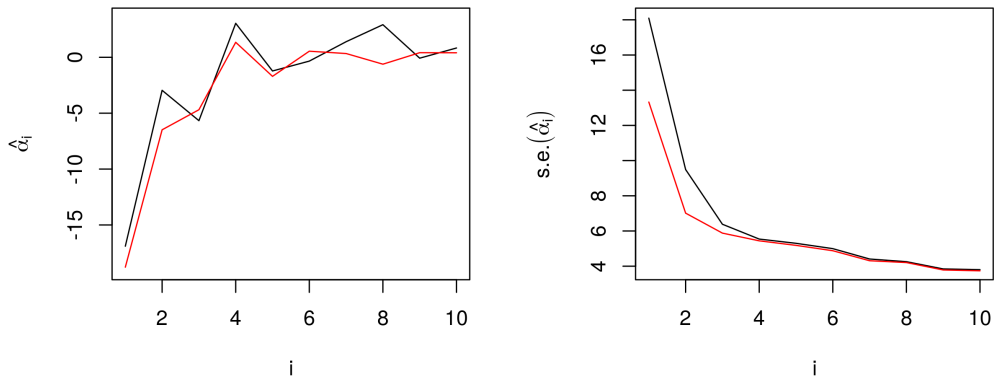

Figure 4: Forecasts (left) and prediction standard errors (right) for the EOF coefficients in October 1997 using a lag-6 time-series model estimated using the method of moments (black) and a lag-1 state-space model estimated using the EM algorithm (red).

```
        DSTM_Results$Ceta
}
```

It is instructive to compare the predictions and the prediction standard errors of the forecasted EOF coefficients using the two models; see Figure 4. While the prediction standard errors for the state-space model are slightly lower (which is expected since a measurement-error component of variability has been filtered out), it is remarkable that forecasts from the lag-6 time-series model are quite similar to those of the lag-1 state-space model. These

two models and their respective inferences can be expected to differ when there is more nonlinearity in the process and/or the data are less complete in space and time.

```
## pdf
##   2
```

Bibliography

Schott, J. R. (2017), *Matrix Analysis for Statistics*, 3rd ed., Hoboken, NJ: John Wiley & Sons.