MORE DETAIL ON FINDING UNCONDITIONAL DISTRIBUTION OF A STATIONARY AR

1. Why these notes?

- This topic was discussed in notes we went over in class and that are on the course web site (as "Notes on linear time series models").
- However, Liyu observed that in the last exercise a significant fraction of the class did not seem to konw how to find the unconditional distribution.
- Here we reproduce the part of the previous notes that describes the linear-solver method.
- In addition, we describe another easily implemented iterative method, the "doubling algorithm", that may work better on large systems.

2. $E[y]$ and $R_y(0)$ for a stationary AR: the previous slides, for review

- Whereas for an MA model computing the full R_y function is straightforward polynomial matrix multiplication, finding R_y is more work for an AR model.
- In the first-order case, once we know $R_y(0)$, we can find $R_y(s) = B^s R_y(0)$ for all s. But for $R_y(0)$ we need to solve

$$R_y(0) = B\, R_y(0) B' + \Sigma_\varepsilon \,,$$

which is a system of linear equations in the elements of $R_y(0)$.

3. $E[y]$ and $R_y(0)$ for a stationary AR

- It's a big system, and there's a literature on ways to solve it speedily. One name for it is "Lyapunov equation".
- For small models,

$$(I - B \otimes B)\overrightarrow{R_y(0)} = \overrightarrow{\Sigma_\varepsilon} \,,$$

where the arrows indicate stacking of the columns of a matrix to form a vector, can be solved. Because $R_y(0)$ is symmetric the system as written above has more equations than unknowns, so some rows of the system might have to be dropped to solve it

4. $E[y]$ and $R_y(0)$ for a stationary AR

- If the system has a constant term c, the mean $\bar{y}$ of the stationary process is found by solving

$$\bar{y} = B\bar{y} + c \Rightarrow \bar{y} = (I - B)^{-1} c \,.$$

Date: May 18, 2018.

5. The doubling algorithm

By successive substitution we can derive

$$R_y(0) = \Sigma_\varepsilon + B\,R_y(0)B' = \Sigma_\varepsilon + B\Sigma_\varepsilon B' + B^2 R_y(0)(B')^2 = \sum_{s=0}^{\infty} B^s \Sigma_\varepsilon (B')^s \,. \tag{1}$$

Note that if

$$\Omega_J = \sum_{s=0}^{2^J-1} B^s \Sigma_\varepsilon (B')^s \,,$$

then

$$\Omega_{J+1} = \Omega_J + B^{2^J} \Omega_J (B')^{2^J} \,.$$

From this expression, we can see that the following iterative procedure will calculate $R_y(0)$:

```
doubling <- function(B, Sigma) {
    Omega <- Sigma
    C <- B
    nit <- 1
    delta <- 10
    while (delta > 1e-7 && nit < 100) {
        nit <- nit + 1
        diff <- C %*% Omega %*% t(C)
        Omega <- Omega + diff
        delta <- sum(abs(diff))
        C <- C %*% C
    }
    return(list(Omega=Omega, nit=nit))
}
```

6. Application to exercise

In the exercise, you needed to calculate an unconditional distribution for z_1, a 2×1 vector that evolves according to a stationary, first order, AR. It thus fits exactly into the framework of the previous slides, with A in the exercise playing the role of B in the notes. So your "explain in enough detail" task should have included a description how to program calculation of $R_y(0)$ either by the Kronecker product linear system or (if you checked another reference) a doubling algorithm or an eigenvalue-decomposition-based Lyapunove equation solver (which we have not discussed).

Note that all these methods require that all the eigenvalues of B be less than one in absolute value, so that the model implies stationarity. Numerical stability issues can arise for any of the methods if B has eigenvalues close to one in absolute value.