Spencer Lyon

Sargent RA: M.O. 2

Due on Sunday, July 28, 2013

August 8, 2013

Problem 2.6

Consider the stochastic process $\{c_t, z_t\}$ defined by equations (1) in exercise 2.5. Assume the parameter values described in part b, item i. If possible, assume the initial conditions are such that $\{c_t, z_t\}$ is covariance stationary.

a. Compute the initial mean and covariance matrix that make the process covariance stationary.

b. For the initial conditions in part a, compute numerical values of the following population linear regression:

$$c_{t+2} = \alpha_0 + \alpha_1 z_t + \alpha_2 z_{t-4} + w_t$$

where $\begin{bmatrix} 1 & z_t & z_{t-4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$

a. As noted in my solution to problem 5, the system described there can be put in to the canonical form for a stochastic linear difference equation as follows:

$$\begin{array}{ccccc} x_{t+1} = & & Ax_t + & & Cw_{t+1} \\ \begin{bmatrix} c_{t+1} \\ c_t \\ z_{t+1} \\ z_t \\ 1 \end{bmatrix} & = & \begin{bmatrix} \rho_1 & \rho_2 & \delta_1 & \delta_2 & \alpha_c \\ 1 & 0 & 0 & 0 & 0 \\ \gamma_1 & \gamma_2 & \phi_1 & \phi_2 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_t \\ c_{t-1} \\ z_t \\ z_{t-1} \\ 1 \end{bmatrix} & + & \begin{bmatrix} \psi_1 & 0 \\ 0 & 0 \\ 0 & \psi_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_{1,t+1} \\ w_{2,t+1} \end{bmatrix} \end{array}$$

Using this form, the covariance stationary mean is found by computing the eigenvector of A that corresponds to the single unit eigenvalue of A. I did this using the included python program and got that

$$\mu = \begin{bmatrix} 2 & 2 & 0 & 0 & 1 \end{bmatrix}$$

So solve for the covariance stationary covariance matrix, I simply used the command `doublej(A, C.dot(C.T))`. The matrix I got for C_x is

$$\begin{bmatrix} 1.970 & 1.249 & 0.241 & 0.487 & 0.000 \\ 1.249 & 1.970 & 0.071 & 0.241 & 0.000 \\ 0.241 & 0.071 & 1.579 & 0.921 & 0.000 \\ 0.487 & 0.241 & 0.921 & 1.579 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \end{bmatrix}$$

b. To solve for the coefficients in this problem I will use equation 2.5.3, which states

$$\beta = (EYX')[E(XX')]^{-1}$$

where $X' = [\,1 \; z_t \; z_{t-4}\,]$ and $Y = c_{t+2}$. This allows me to write the following expression for $E(XX')$:

$$E(XX') = E \begin{bmatrix} 1 & z_t & z_{t-4} \\ z_t & z_t^2 & z_{t-4}z_t \\ z_{t-4} & z_{t-4}z_t & z_{t-4}^2 \end{bmatrix}$$

I can now evaluate what these expressions are by passing the expectation through and noting from part a that $E[z_t] = 0 \forall t$. This allows me to compute directly the first row and column of the matrix. To compute the rest of it I turn to the definition of covariance: $\text{cov}(x, y) = E(x - \mu_x)E(y - \mu_y)'$. Applying this definition and recalling that $\mu_z = 0$, I can simplify the matrix above in the following manner:

$$E(XX') = E \begin{bmatrix} 1 & z_t & z_{t-4} \\ z_t & z_t^2 & z_{t-4}z_t \\ z_{t-4} & z_{t-4}z_t & z_{t-4}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{var}(z_t) & \text{cov}(z_{t-4}, z_t) \\ 0 & \text{cov}(z_{t-4}, z_t) & \text{var}(z_{t-4}) \end{bmatrix}$$

The expressions above can be found in certain entries of $C_x(j)$:

- $\text{var}(z_t)$ is the 3, 3 entry of $C_x(0)$
- $\text{cov}(z_{t-4}, z_t)$ is the 3, 3 entry of $C_x(-4) = \left[A^4 C_x(0)\right]'$
- $\text{var}(z_t)$ is the 3, 3 entry of $C_x(0)$
- $\text{var}(z_{t-4})$ is also the 3, 3 entry of $C_x(0)$ because the distribution is time-invariant (covariance stationary)

Now to evaluate the other expectation.

$$E(YX') = E \begin{bmatrix} c_{t+2} \\ c_{t+2}z_t' \\ c_{t+2}z_{t-4}' \end{bmatrix}$$

The first row of this vector is easy and obvious. The other two rows can be computed directly by expanding out the definition of the covariance. I will only show this for the second row, but the third follows similarly.

$$
\begin{aligned}
\operatorname{cov}(c_{t+2}, z_t) &= E(c_{t+2} - \mu_c) E(z_t - \mu_z)' \\
&= E(c_{t+2} z_t') + E(c_{t+2}(-\mu_z)') + E((-\mu_c) z_t') + E((-\mu_c)(-\mu_z)') \\
&= E(c_{t+2} z_t') + E(c_{t+2})(-0)' + E((2)0') + E((2)(0)') \\
&= E(c_{t+2} z_t')
\end{aligned}
$$

which is the expression in the second row of $E(YX')$. This means that we can say the following two things:

- $E(c_{t+2} z_t') = \operatorname{cov}(c_{t+2}, z_t)$, which is the $1,3$ element of $C_x(2)$
- $E(c_{t+2} z_{t-4}') = \operatorname{cov}(c_{t+2}, z_{t-4})$, which is the $1,4$ element of $C_x(5)$

I am now finally ready to evaluate the expressions in each matrix, take the product of them, and obtain an expression for β.

$$
\begin{aligned}
\beta &= (EYX')[E(XX')]^{-1} \\
&= \begin{bmatrix} 2.000 \\ 0.502 \\ -0.050 \end{bmatrix} \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 1.579 & -0.034 \\ 0.000 & -0.034 & 1.579 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} 2.000 \\ 0.317 \\ -0.025 \end{bmatrix}
\end{aligned}
$$

```python
from __future__ import division
import sys
import numpy as np
from scipy.linalg import inv, eig
import pandas as pd
from rmt_utils import doublej
from matrix2latex import matrix2latex as to_tex

bmat_tex = lambda i: to_tex(i, None, 'bmatrix', formatColumn=['%.3f'] * 15)


class StochasticLinearDiff(object):
    """
    Represents and computes various things for a model in the form
    of the canonical stochastic linear difference equation:

    .. math::

        x_{t+1} = A x_t + C w_{t+1}
    """

    def __init__(self, A, C):
        self.A = A
        self.C = C

        # Evaluate eigenvalues and vectors for use later on. Check boundedness
        evals, evecs = eig(self.A, left=False, right=True)
        self.evals, self.evecs = evals, evecs
        self.unbounded = np.abs(evals).max() > 1

    def Cx(self, j=0):
        "Covariance stationary covariance matrix"
        if not self.unbounded:
            c_x = doublej(self.A, self.C.dot(self.C.T))

            # Return if we want C_x(0)
```

Problem 2.6 continued on next page...

```
            if j == 0:
                return c_x
            else:
                # Or evaluate C_x(abs(j))
                c_xj = np.linalg.matrix_power(self.A, abs(j)).dot(c_x)
            if j < 0:
                return c_xj.T # transpose if j < 0
            else:
                return c_xj

        else:
            msg = 'This computation will not work because the eigenvalues'
            msg += '\nof A are not all below 1 in modulus.'
            raise ValueError(msg)

    @property
    def mu(self):
        "Covariance stationary mean"
        if self.unbounded:
            msg = 'This computation will not work because the eigenvalues {0}'
            msg += '\nof A are not all below 1 in modulus.'
            raise ValueError(msg.format(self.evals))

        # Try to get index of unit eigenvalue
        try:
            ind = np.where(self.evals == 1)[0][0]
        except IndexError:
            raise ValueError("The A matrix doesn't have any unit eigenvalues")

        # compute Stationary mean using the eigenvector for unit eigenvalue
        return self.evecs[:, ind] / self.evecs[-1, ind]
case5_2 = ([0.8, -0.3], 1., [0.2, 0], [0., 0.], [0.7, -0.2], 2., 1.)


def p2_6(rho, alpha, delta, gamma, phi, psi1, psi2, verbose=False):
    """
    Solution to problem 2.5 part b for RMT4

    Problem Text
    ============

    TODO: Fill this in

    Parameters
    ==========
    rho : array_like, dtype=float
        The value of rho from the problem description

    alpha : float
        The value of alpha from the problem description

    delta : array_like, dtype=float
        The value of delta from the problem description

    gamma : array_like, dtype=float
        The value of gamma from the problem description

    phi : array_like, dtype=float
        The value of phi from the problem description

    psi1 : float
        The value of psi1 from the problem description

    psi2 : float
        The value of psi2 from the problem description

    """
    A = np.array([
                [rho[0], rho[1], delta[0], delta[1], alpha],
                [1, 0, 0, 0, 0],
                [gamma[0], gamma[1], phi[0], phi[1], 0],
                [0, 0, 1, 0, 0],
                [0, 0, 0, 0, 1]
                ], dtype=float)

    C = np.array([[psi1, 0],
                 [0, 0],
                 [0, psi2],
                 [0, 0],
                 [0, 0]
```

Problem 2.6 continued on next page. . .

```
                 ], dtype=float)

    # part A
    diff_eq = StochasticLinearDiff(A, C)

    cx = diff_eq.Cx()
    mu = diff_eq.mu

    # part b
    e_xx = np.array([[1., 0., 0.],
                     [0., cx[2, 2], diff_eq.Cx(4)[2, 2]],
                     [0., diff_eq.Cx(-4)[2, 2], cx[2, 2]]])

    e_xy = np.array([2, diff_eq.Cx(2)[0, 2], diff_eq.Cx(5)[0, 3]])

    beta = e_xy.dot(inv(e_xx))

    if verbose:
        print('\n\nPROBLEM 2.6 %s\n\n' % ('#' * 65))
        vals = "C_X(0) = {0}\nmu = {1}\nE(XX') = {2}\nE(YX') = {3}\nbeta= {4}"
        print(vals.format(bmat_tex(cx),
                          bmat_tex(np.abs(mu)),
                          bmat_tex(e_xx),
                          bmat_tex(e_xy),
                          bmat_tex(beta)))

    return cx, mu, beta

cx, mu, beta = p2_6(*case5_2, verbose=False)
```

□

Problem 2.10

Let P be a transition matrix for a Markov chain. Suppose that P' has two distinct eigenvectors π_1, π_2 corresponding to unit eigenvalues of P' . Scale π_1 and π_2 so that they are vectors of probabilities (i.e., elements are nonnegative and sum to unity). Prove for any $\alpha \in [0,1]$ that $\alpha\pi_1 + (1-\alpha)\pi_2$ is an invariant function of P .

I am given that π_1 and π_2 are the only eigenvectors corresponding to the unit eigenvalues of P. Any convex combination of eigenvectors corresponding to the same eigenvalue is also an eigenvector corresponding to that same eigenvector. The quantity given in the problem $(\alpha\pi_1 + (1-\alpha)\pi_2)$ is a convex combination, which means that it is also an eigenvector corresponding to a unit eigenvalue of P.

By definition, the eigenvectors of a Markov transition matrix P corresponding to unit eigenvalues are invariant functions of P. □

Problem 2.14

Consider a Markov chain with transition matrix

$$\begin{bmatrix} .5 & .5 & 0 & 0 \\ .1 & .9 & 0 & 0 \\ 0 & 0 & .9 & .1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with state space $X = \{e_i, i = 1, \ldots, 4\}$ where e_i is the ith unit vector. A random variable y_t is a function $y_t = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} x_t$ of the underlying state.

a. Find all stationary distributions of the Markov chain.

b. Can you find a stationary distribution for which the Markov chain ergodic?

c. Compute all possible limiting values of the sample mean $\frac{1}{T}\sum_{t=0}^{T-1} y_t$ as $T \to \infty$

a. The stationary distributions of the matrix P are the left eigenvectors associated with unit eigenvalues. Doing this computation for the given transition matrix yields the following stationary distributions:

$$\pi_1 = \begin{bmatrix} 1/6 & 5/6 & 0 & 0 \end{bmatrix}$$
$$\pi_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

Any convex combination of π_1 and π_2 is also a stationary distribution of P.

b. To answer this question, I first need to find the invariant functions associated with the chain P. These are the right eigenvectors of P and I found them to be:

$$\bar{y}_1 = \begin{bmatrix} .5 & .5 & 0 & 0 \end{bmatrix}$$
$$\bar{y}_2 = \begin{bmatrix} 0 & 0 & .5 & .5 \end{bmatrix}$$

Note that eigenvectors are only constant up to a scalar multiple, so it is fair to write them in terms of a scalar multiple, like so (Note that α and β can be any real number):

$$\bar{y}_1 = \begin{bmatrix} \alpha & \alpha & 0 & 0 \end{bmatrix}$$
$$\bar{y}_2 = \begin{bmatrix} 0 & 0 & \beta & \beta \end{bmatrix}$$

Now, any convex combination of the eigenvectors of a matrix is also an eigenvector of the same matrix. That means we have a continuum of stationary distributions of the form:

$$\gamma\pi_1 + (1-\gamma)\pi_2$$
$$\gamma \begin{bmatrix} 1/6 & 5/6 & 0 & 0 \end{bmatrix} + (1-\gamma)\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\gamma \in [0,1]$. There are 3 cases that we will examine separately:

1. $\gamma \in (0,1)$: In this situation, there is positive probability on elements 1,2, and 4 of the stationary distribution. This means that for the chain to be ergodic under these circumstances, elements 1,2, and 4 of the invariant functions must be equal. We see that they are not. Therefore the chain is not ergodic when $\gamma \in (0,1)$.
2. $\gamma = 0$: Here the stationary distribution is π_1 and there is positive probability in elements 1 and 2. We also see that the invariant functions all have equal values for elements 1 and 2, therefore the chain is ergodic under π_1.

3. $\gamma = 1$: This time the stationary distribution is π_1 and there is positive probability only in element 4. All invariant functions have a constant scalar in as the 4th element, so the chain is also ergodic under π_2.

c. The value of the sample mean will depend on the initial distribution π_0. However, regardless of the initial distribution, π_t will eventually converge to one of the ergodic distributions and the sample mean will converge to the value implied by the ergodic distribution. In part b I showed that the only two ergodic distributions were π_1 and π_2. This means that the limiting values of the sample mean are simply the average of $\bar{y}$ weighted by the elements in π_1 and π_2. Those values are below.

- Under π_1: 1.83334
- Under π_2: 4.0

```
from __future__ import division
import sys
import numpy as np
from scipy.linalg import inv, eig
import pandas as pd
from rmt_utils import doublej
from matrix2latex import matrix2latex as to_tex

bmat_tex = lambda i: to_tex(i, None, 'bmatrix', formatColumn=['%.3f'] * 15)


class Markov(object):
    """
    Do basic things with Markov matrices.
    """

    def __init__(self, P, verbose=False):
        self.P = P
        self.verbose = verbose

    def __repr__(self):
        msg = "Markov process with transition matrix P = \n{0}"
        return msg.format(self.P)

    def stationary_distributions(self):
        evals, l_evecs, r_evecs = eig(self.P, left=True, right=True)
        self.evals, self.l_evecs, self.r_evecs = evals, l_evecs, r_evecs
        units = np.where(evals == 1)[0]
        stationary = []
        for i, ind in enumerate(units):
            sd_name = 'sd{0}'.format(i + 1)
            sd_vec = l_evecs[:, ind]

            # Normalize to be probability vector
            sd_vec = sd_vec * (-1) if all(sd_vec <= 0) else sd_vec
            sd_vec /= sd_vec.sum()
            self.__setattr__(sd_name, sd_vec)
            stationary.append(sd_vec)
            if self.verbose:
                msg = 'Set instance variable %s for stationary distribution'
                print(msg % sd_name)
        return stationary

    def invariant_distributions(self):
        units = np.where(self.evals == 1)[0]
        invariant = []
        for i, ind in enumerate(units):
            id_name = 'id{0}'.format(i + 1)
            id_vec = self.r_evecs[:, ind]

            # Normalize to be probability vector
            id_vec = id_vec * (-1) if all(id_vec <= 0) else id_vec
            id_vec /= id_vec.sum()
            self.__setattr__(id_name, id_vec)
            invariant.append(id_vec)
            if self.verbose:
                msg = 'Set instance variable %s for invariant distribution'
                print(msg % id_name)
```

```
        return invariant

P_14 = np.array([[.5, .5, .0, .0],
                 [.1, .9, .0, .0],
                 [.0, .0, .9, .1],
                 [.0, .0, .0, 1]])


def p2_14(P=P_14, verbose=False):
    """
    Solves for stationary distributions and long-run averages for
    :math:`y_t = \\bar{y} x_t = [1 2 3 4] x_t` given a transition matrix
    P

    Parameters
    ==========
    P : array_like, dtype=float
        The 2d numpy array representing the Markov transition matrix

    Returns
    =======
    markov : Markov
        The object of type Markov representing the process
    stationary : array_like, dtype=float
        The list of arrays representing stationary distributions of the
        process P
    invariant : array_like, dtype=float
        The list of arrays representing the invariant distributions of
        the process P
    lr_means : array_like, dtype=float
        The long-run mean of the process under each stationary
        distribution

    """
    markov = Markov(P)
    stationary = markov.stationary_distributions()
    invariant = markov.invariant_distributions()

    ybar = np.array([1, 2, 3, 4])

    # Long run mean values
    lr_means = [markov.sd1.dot(ybar), markov.sd2.dot(ybar)]

    if verbose:
        print('\n\nPROBLEM 2.14 %s\n\n' % ('#' * 65))
        vals = "\\pi_0 = {0}\n\\pi_1 = {1}"
        print(vals.format(bmat_tex(stationary[0]), bmat_tex(stationary[1])))

    return markov, stationary, invariant, lr_means


markov_14, _, _, means = p2_14(verbose=False)
```

□

Problem 2.17 – Lake model

A worker can be in one of two states, state 1 (unemployed) or state 2 (employed). At the beginning of each period, a previously unemployed worker has probability $\lambda = \int_{\bar{w}}^{B} dF(w)$ of becoming employed. Here $\bar{w}$ is his reservation wage and $F(w)$ is the c.d.f. of a wage offer distribution. We assume that $F(0) = 0, F(B) = 1$. At the beginning of each period an unemployed worker draws one and only one wage offer from F. Successive draws from F are i.i.d. The worker's decision rule is to accept the job if $w \geq \bar{w}$, and otherwise to reject it and remain unemployed one more period. Assume that $\bar{w}$ is such that $\lambda \in (0,1)$. At the beginning of each period, a previously employed worker is fired with probability $\delta \in (0,1)$. Newly fired workers must remain unemployed for one period before drawing a new wage offer.

a. Let the state space be $X = \{e_i, i = 1,2\}$, where e_i is the ith unit vector. Describe the Markov chain on X that is induced by the description above. Compute all stationary distributions of the chain. Under what stationary distributions, if any, is the chain ergodic?

b. Suppose that $\lambda = .05, \delta = .25$. Compute a stationary distribution. Compute the fraction of his life that an infinitely lived worker would spend unemployed.

c. Drawing the initial state from the stationary distribution, compute the joint distribution $g_{ij} = \text{Prob}(x_t = e_i, x_{t-1} = e_j)$ for $i = 1,2, j = 1,2$.

d. Define an indicator function by letting $I_{ij,t} = 1$ if $x_t = e_i, x_{t-1} = e_j$ at time t and 0 otherwise. Compute

$$\lim_{t\to\infty} \frac{1}{T} \sum_{t=1}^{T} I_{ij,t}$$

for all four i, j combinations.

e. Building on your results in part d, construct method of moments estimators of λ and δ. Assuming that you know the wage offer distribution F, construct a method of moments estimator of the reservation wage $\bar{w}$.

f. Compute maximum likelihood estimators of λ and δ.

g. Compare the estimators you derived in parts e and f.

h. *Extra Credit.* Compute the asymptotic covariance matrix of the maximum likelihood estimators of λ and δ.

a. The state space can be represented by the transition matrix:

$$P = \begin{bmatrix} 1-\lambda & \lambda \\ \delta & 1-\delta \end{bmatrix}$$

The stationary distributions of this matrix are the left eigenvectors corresponding to the unit eigenvalues. In this case the eigenvalues are equal to $\psi_1 = 1, \psi_2 = 1 - \delta - \lambda$. The left eigenvector corresponding to the unit eigenvalue is given as (note I normalize so it sums to 1):

$$\pi_0 = \begin{bmatrix} 1 \\ \delta/\lambda \end{bmatrix} \to \begin{bmatrix} \frac{\lambda}{\delta+\lambda} \\ \frac{\delta}{\delta+\lambda} \end{bmatrix}$$

To determine the ergodicity of this stationary distribution, I need to evaluate the invariant functions of the Markov chain. The invariant functions are the right eigenvectors corresponding to the unit eigenvalues. As noted above, there is a single unit eigenvalue for this process and it has a corresponding right eigenvector equal to

$$\xi = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix}$$

This chain is ergodic because both elements of π_0 are positive and both elements of ξ are equal.

b. When $\lambda = 0.05$ and $\delta = 0.25$ the stationary distribution is

$$\pi_0 = \begin{bmatrix} 5/6 \\ 1/6 \end{bmatrix}$$

This means that the infinitely lived worker would expect to be un-employed for $5/6 \approx 83.33\%$ of his life.

c. In this problem, the values of g_{ij} are given by equation 2.3.1, which in this case means

$$g_{ij} = \text{Prob}(x_t = e_i, x_{t-1} = e_j) = \text{Prob}\left(x_t = e_i | x_{t-1} = e_j\right) \text{Prob}\left(x_{t-1} = e_j\right)$$

In this case $\text{Prob}\left(x_t = e_i | x_{t-1} = e_j\right)$ is simply the ji element of P and $\text{Prob}\left(x_{t-1} = e_j\right)$ is the j element of π_0. Doing the computation yields

- $g_{11} = P_{11}\pi_{0,1} = (1-\lambda)\pi_{0,1} = \frac{19}{20}\frac{5}{6} = \frac{19}{24}$
- $g_{12} = P_{21}\pi_{0,1} = (\delta)\pi_{0,2} = \frac{1}{4}\frac{1}{6} = \frac{1}{24}$
- $g_{21} = P_{12}\pi_{0,2} = (\lambda)\pi_{0,1} = \frac{1}{20}\frac{5}{6} = \frac{1}{24}$
- $g_{22} = P_{22}\pi_{0,2} = (1-\delta)\pi_{0,2} = \frac{3}{4}\frac{1}{6} = \frac{1}{8}$

d. In this problem I was asked to find the average value of an indicator function. The way the indicator function was defined makes this average equal to the expected value of I_{ij}. The expected value of an indicator function for a particular state is simply the probability of being in the particular state. Thus, I can say

$$\lim_{t\to\infty} \frac{1}{T}\sum_{t=1}^{T} I_{ij,t} = E\left[I_{ij}\right] = P(ij) = g_{ij}$$

where g_{ij} are the values from part c. The answer is

- $I_{11} = g_{11} = (1-\lambda)\pi_{0,1} = \frac{19}{24}$
- $I_{12} = g_{12} = (\delta)\pi_{0,2} = \frac{1}{24}$
- $I_{21} = g_{21} = (\lambda)\pi_{0,1} = \frac{1}{24}$
- $I_{22} = g_{22} = (1-\delta)\pi_{0,2} = \frac{1}{8}$

e. Let $\hat{I}_{ij}$ be the sample mean corresponding to the theoretical means calculated in the previous part. I will construct a method of moments estimator for λ by letting $\frac{\hat{I}_{11}}{\hat{I}_{12}}$ equal to the equivalent ratio from the above. This will allow me to write an expression for $\hat{\lambda}$ (the MOM estimator of λ) only in terms of the sample moments on $\hat{I}_{ij}$ as follows.

$$\begin{aligned}
\frac{\hat{I}_{11}}{\hat{I}_{21}} &= \frac{\lim_{t\to\infty}\frac{1}{T}\sum_{t=1}^{T} I_{11,t}}{\lim_{t\to\infty}\frac{1}{T}\sum_{t=1}^{T} I_{21,t}} \\
&= \frac{(1-\hat{\lambda})\pi_{0,1}}{(\hat{\lambda})\pi_{0,1}} \\
&= \frac{1-\hat{\lambda}}{\hat{\lambda}} \\
\hat{\lambda} &= \frac{1}{\frac{\hat{I}_{11}}{\hat{I}_{21}}+1} = \frac{\hat{I}_{12}}{\hat{I}_{11}+\hat{I}_{12}}
\end{aligned}$$

Repeating a similar process for δ gives the following:

$$\begin{aligned}
\frac{\hat{I}_{22}}{\hat{I}_{12}} &= \frac{\lim_{t\to\infty}\frac{1}{T}\sum_{t=1}^{T} I_{22,t}}{\lim_{t\to\infty}\frac{1}{T}\sum_{t=1}^{T} I_{12,t}} \\
&= \frac{(1-\hat{\delta})\pi_{0,1}}{(\hat{\delta})\pi_{0,1}} \\
&= \frac{1-\hat{\delta}}{\hat{\delta}} \\
\hat{\delta} &= \frac{1}{\frac{\hat{I}_{22}}{\hat{I}_{12}}+1} = \frac{\hat{I}_{21}}{\hat{I}_{21}+\hat{I}_{22}}
\end{aligned}$$

To find a method of moments estimator for $\bar{x}$ (call is $\hat{\bar{w}}$), I will return to the definition of λ and replace it with $\hat{\lambda}$ as such

$$\hat{\lambda} = \int_{\bar{w}}^{B} dF(w)$$

Evaluating the integral (keeping in mind that $F(B)=1$), I can solve for an expression for $\hat{\bar{w}}$. Note that I define F^{-1} to be the inverse of the given $F(w)$ in the sense that if $F(x)=y$ then $F^{-1}(y)=x$.

$$\begin{aligned}
\hat{\lambda} &= \int_{\bar{w}}^{B} dF(w) \\
\hat{\lambda} &= F(B) - F(\hat{\bar{w}}) \\
\hat{\lambda} &= 1 - F(\hat{\bar{w}}) \\
\hat{\bar{w}} &= F^{-1}(1-\hat{\lambda})
\end{aligned}$$

f. To compute the maximum likelihood estimators of λ and δ, I begin with the likelihood function given in section 2.2.9:

$$L = \pi_{0,i_0}\prod_i\prod_j P_{ij}^{n_{ij}}$$

where n_{ij} the number of times a transition from state i to state j occurs in a sample data set. To obtain maximum likelihood estimators I write the likelihood function in terms of the free parameters λ and δ and choose those parameters so that the likelihood function is maximized. It is often easier to do this when working with the log-likilihood function, which is just the natural logarithm of the likelihood function. Because the logarithm is a monotonic transformation the λ and δ that maximize the log-likelihood function will also maximize the likelihood function. The log-likelihood function is:

$$l = \log(L) = \log(\pi_{0,i_0}) + \sum_i\sum_j n_{ij}\log(P_{ij})$$

Taking the derivative of this function with respect to both parameters yields:

$$\frac{\partial l}{\partial \lambda} = \sum_i \sum_j n_{ij} \frac{\partial P_{ij}/\partial \lambda}{P_{ij}}$$
$$\frac{\partial l}{\partial \delta} = \sum_i \sum_j n_{ij} \frac{\partial P_{ij}/\partial \delta}{P_{ij}}$$

In terms of this problem, with the transition matrix P given from part a, these expressions are evaluated as follows:

$$\begin{aligned}
\frac{\partial l}{\partial \lambda} &= \sum_i \sum_j n_{ij} \frac{\frac{\partial P_{ij}}{\partial \lambda}}{P_{ij}} \\
&= n_{11}\left(\frac{-1}{1-\lambda}\right) + n_{12}\left(\frac{1}{\lambda}\right) + n_{21}(0) + n_{22}(0) \\
\frac{\partial l}{\partial \delta} &= \sum_i \sum_j n_{ij} \frac{\frac{\partial P_{ij}}{\partial \delta}}{P_{ij}} \\
&= n_{11}(0) + n_{12}(0) + n_{21}\left(\frac{1}{\delta}\right) + n_{22}\left(\frac{-1}{1-\delta}\right)
\end{aligned}$$

Finally, to obtain the maximum likelihood parameter estimates, I set these two equations equal to zero and solve them for δ and λ. Doing so yields the following:

$$\tilde{\lambda} = \frac{n_{12}}{n_{11} + n_{12}}$$

$$\tilde{\delta} = \frac{n_{21}}{n_{21} + n_{22}}$$

g. The method of moments and maximum likelihood estimators for λ and δ are actually equivalent. This can be understood by expanding the values $\hat{I}_{ij}$ in the method of expression for the method of moment estimators. To show how to do this, I will outline a numerical algorithm that could be used to estimate $\hat{I}_{ij}$.

- Setup:
  - Obtain, via simulation or otherwise, a sample chain $\{x\}_{t=1}^{T}$ that comes from the transition matrix.
  - Determine what value each state is given in the chain. Express this as a vector V.
- To estimate $\hat{I}_{ij}$
  - Start with `n=0`
  - For `n` $\in [1, T)$
    - Let `a` = x_{t-1} and `b` = x_t
    - If `b ==` V_j `and a ==` V_i, then `n = n+1`
  - Compute $\hat{I}_{ij}$ `= n / (T - 1)`

If we plug this expression into the method of moment estimators, we will see that they end up being equal to the maximum likelihood estimators.

$$
\begin{aligned}
\hat{\lambda} &= \frac{\hat{I}_{12}}{\hat{I}_{11} + \hat{I}_{12}} \\
&= \frac{n_{12}/(T-1)}{n_{11}/(T-1) + n_{12}/(T-1)} \\
&= \frac{n_{12}}{n_{11} + n_{12}} \\
&= \tilde{\lambda}
\end{aligned}
$$

$$
\begin{aligned}
\hat{\delta} &= \frac{\hat{I}_{21}}{\hat{I}_{21} + \hat{I}_{22}} \\
&= \frac{n_{21}/(T-1)}{n_{21}/(T-1) + n_{22}/(T-1)} \\
&= \frac{n_{21}}{n_{21} + n_{22}} \\
&= \tilde{\delta}
\end{aligned}
$$

□

Problem 2.20 – Random Walk

A scalar process x_t follows the process

$$x_{t+1} = x_t + w_{t+1}$$

where w is an iid $N(0,1)$ scalar process and $X_0 \sim N(\hat{x}_0, \Sigma_0)$. Each period, an observer receives two signals in the form of a $2x1$ vector y_t that obeys

$$y_t = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_t + v_t$$

where the $2x1$ process v is iid with distribution $v_t \sim N(0, R)$ where $R = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

a. Suppose that $\Sigma_0 = 1.36602540378444$. For $t \geq 0$, find formulas for $E[x_t | y_{t-1}]$, where y_{t-1} is the history of y_s for s from 0 to $t-1$.

b. Verify numerically that the matrix $A - KG$ in formula (2.9.3) is stable.

c. Find an infinite-order vector autoregression for y_t .

NOTE: I would like to acknowledge that Chase Coleman helped a great deal with this problem.

This problem is set up similar as a system similar to the Kalman filter system defined by equations 2.7.1 and 2.7.2 with $A_0 = C = 1, R = I_{2x2}$, and $G = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, where I is the identity matrix.

a. To begin, I would like to make one definition: $\hat{x}_t = E\left[x_t | y_{t-1}\right]$. With this in mind, I turn to the Kalman filter equations found in 2.7.12. Below I list those 4 equations, make substitutions for this problem, and present a simplified form of each equation:

Problem 2.20 – Random Walk continued on next page. . .

$$
\begin{aligned}
a_t &= y_t - G\hat{x}_t \\
&= y_t - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \hat{x}_t \\
&= \begin{bmatrix} y - \hat{x}_t \\ y - \hat{x}_t \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
K_t &= A_0 \Sigma_t G' (G \Sigma_t G' + R)^{-1} \\
&= 1 \Sigma_t \begin{bmatrix} 1 \\ 1 \end{bmatrix}' \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} \Sigma_t \begin{bmatrix} 1 \\ 1 \end{bmatrix}' + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \\
&= \begin{bmatrix} \frac{\Sigma_t}{2\Sigma_t + 1} & \frac{\Sigma_t}{2\Sigma_t + 1} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\hat{x}_{t+1} &= A_0 \hat{x}_t + K_t a_t \\
&= 1 \hat{x}_t + \begin{bmatrix} \frac{\Sigma_t}{2\Sigma_t + 1} & \frac{\Sigma_t}{2\Sigma_t + 1} \end{bmatrix} \begin{bmatrix} y - \hat{x}_t \\ y - \hat{x}_t \end{bmatrix} \\
&= \frac{\hat{x}_t + 2 y_t \Sigma_t}{2\Sigma_t + 1}
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{t+1} &= CC' + K_t R K_t' + (A_0 - K_t G) \Sigma_t (A_0 K_t G)' \\
&= 1 + K_t \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} K_t' + \left(1 - K_t \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \Sigma_t \left(1 - K_t \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)' \\
&= \frac{3\Sigma_t + 1}{2\Sigma_t + 1}
\end{aligned}
$$

Together these 4 equations specify the value of $\hat{x}_t$.

b. The value of $A - KG$, for this problem, is

$$
\begin{aligned}
A - KG &= 1 - \begin{bmatrix} \frac{\Sigma_t}{2\Sigma_t + 1} & \frac{\Sigma_t}{2\Sigma_t + 1} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= \frac{1}{2\Sigma_t + 1}
\end{aligned}
$$

I used an included python program so simulate the path of Σ_t, starting at $\Sigma_0 = 1.36602540378444$. Using this simulated path, I simulated the path of $A - KG$ and plotted it. I have included this path in Figure 1 and it is easy to see that the path is just constant, and therefore stable.

c. To find an infinite order or time-invariant VAR expression for y_t, I turn to section equation 2.9.3 from section 2.9.2 in RMT4. This equation is

$$y_t = G \sum_{j=0}^{\infty} (A - KG)^j K_{y_{t-j-1}} + a_t$$

Where

- $G = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $(A - KG) = \frac{1}{2\Sigma_t + 1}$
- $K_t = \begin{bmatrix} \frac{\Sigma_t}{2\Sigma_t+1} & \frac{\Sigma_t}{2\Sigma_t+1} \end{bmatrix}$
- $a_t = y_t - G\hat{x}_t$

```
from __future__ import division
import sys
import numpy as np
from scipy.linalg import inv, eig
import pandas as pd
from rmt_utils import doublej
from matrix2latex import matrix2latex as to_tex


def p2_20(sigma_0=1.36602540378444, t=5000, plot=True, save_fig=True):
    sigma = np.zeros(t)
    sigma[0] = sigma_0
    for i in range(1, t):
        sigma[i] = (3 * sigma[i - 1] + 1) / (2 * sigma[i - 1] + 1)

    a_gk = 1 / (2 * sigma + 1)
    if plot:
        import matplotlib.pyplot as plt
        fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.plot(range(t), a_gk, 'k--')
        ax.set_title(r'Stability of $A - K_t G$')
        ax.set_ylabel(r'$A - K_t G$')
        ax.set_xlabel('Time (t)')
        if save_fig:
            plt.savefig('./Ch2/p2_20b.eps', format='eps', dpi=1000)
        plt.show()

    return a_gk.max()
```

□

Problem 2.24

A pair of scalar stochastic processes (z_t, y_t) evolves according to the state system for $t \geq 0$:

$$z_{t+1} = 0.9 z_t + w_{t+1}$$
$$y_t = z_t + v_t$$

where w_{t+1} and v_t are mutually uncorrelated scalar Gaussian random variables with means of 0 and variances of 1. Furthermore, $E w_{t+1} v_s = 0$ for all t, s pairs. In addition, $z_0 \sim N(\hat{z}_0, \Sigma_0)$.

a. Is $\{z_t\}$ Markov? Explain

b. Is $\{y_t\}$ Markov? Explain

c. Define what it would mean for the scalar process $\{z_t\}$ to be *covariance stationary*.

d. Find values of $(\hat{z}_0, \Sigma_0)$ that make the process for $\{z_t\}$ covariance stationary.

Problem 2.24 continued on next page. . .

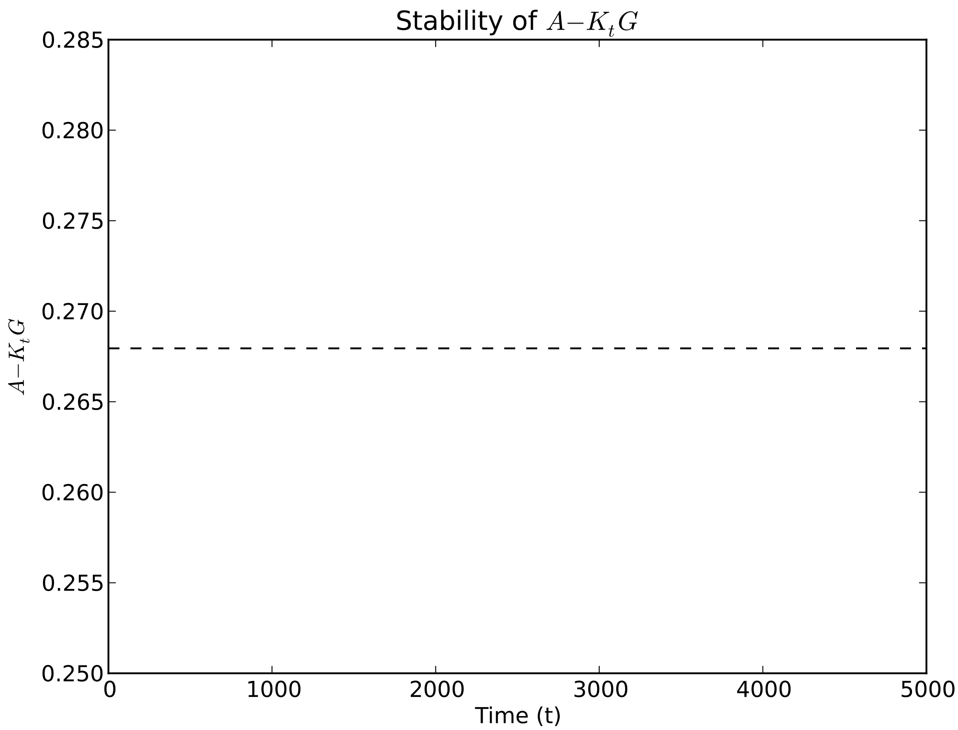

Figure 1: Path of $A - KG$ for random walk process starting with $\Sigma_0 =$ 1.36602540378444

e. Assume that y_t is observable, but that z_t is not. Define what it would mean for the scalar process y_t to be *covariance stationary*.

f. Describe in as much detail as you can how to represent the distribution of y_t conditional on the infinite history y_{t-1} in the form $y_t \sim N(E[y_t|y_{t-1}], \Omega_t)$.

a. Yes, $\{z_t\}$ is Markov because it satisfies the property that $\text{Prob}(z_t|z_{t-1} \dots z_{t-k}) = \text{Prob}(z_t|z_{t-1})$.

b. I believe that y_t is also Markov. I look at the expected value of x and y to argue my case.

$$E[y_t] = E[z_t + v_t] = E[0.9z_{t-1} + w_{t+1}] + 0 = 0.9z_{t-1}$$

$$E[z_t] = E[0.9z_{t-1} + w_{t+1}] = 0.9z_{t-1} + 0$$

Therefore, I can say that $E[y_t] = E[z_t]$ and I have already explained how z_t is Markov.

c. z_t being covariance stationary would mean two things:

1. The expected value of z_t is not a function of t. In other words

$$E[z_t] = E[z_0] = \mu_z$$

2. The sequence of auto-covariance matrices $E\left(z_{t+j} - Ez_{t+j}\right)(z_t - Ez_t)'$ depend only on the separation dates j and not the time period t.

d. To find the values of $\hat{z}_0$ and Σ_0 are found using the equations in a table from section 2.4.2. I repeat the necessary formulas here. Note that $\hat{z}_0 = \mu$ and $C_x(0) = \Sigma_0$.

$$(I - A_0)\mu = 0 \Longrightarrow (1 - 0.9)\hat{z}_0 = 0 \Longrightarrow \hat{z}_0 = 0$$

$$C_x(0) = A_0 C_x(0) A_0' + CC' \Longrightarrow \Sigma_0 = .9\Sigma_0 .9 + 1 \Longrightarrow \Sigma_0 = 5.2632$$

e. If y_t were covariance stationary, the same two properties discussed in part c. would apply. However, these properties would have deeper implications. I will discuss them one at a time.

1. The expected value of y_t is not a function of t. In part b. I explained how $E[y_t] = E[z_t]$. So for property 1 to hold for y_t, it would also hold for z_t.
2. The sequence of auto-covariance matrices $E\left(y_{t+j} - Ey_{t+j}\right)\left(y_t - Ey_t\right)'$ depend only on the separation dates j and not the time period t. I will work directly with the expression for the auto-covariance matrix, do some algebra, substitute $y_t = z_t + v_t$, and do some more algebra below. Note that I use the property that y_t has a time-independent mean going from line 1 to line 2 below.

$$\begin{aligned}
&E\left(y_{t+j} - Ey_{t+j}\right)\left(y_t - Ey_t\right)' \\
&E(y_{t+j} y_t) - \mu_y^2 \\
&E([z_{t+j} + v_{t+j}][z_t + v_t]) - \mu_z^2 \\
&E[z_{t+j} z_t] + E[z_{t+j} v_t] + E[v_{t+j} z_t] + E[v_{t+j} v_t] - \mu_z^2 \\
&E[z_{t+j} z_t] + \text{cov}(z_{t+j}, v_t) + \text{cov}(v_{t+j}, z_t) + \text{cov}(z_{t+j}, v_t) - \mu_z^2 \\
&E[z_{t+j} z_t] - \mu_z^2 \\
&E\left(z_{t+j} - Ez_{t+j}\right)(z_t - Ez_t)'
\end{aligned}$$

So [a], if y_t is covariance stationary, then the unobserved z_t is also covariance stationary.

f. A very useful theorem from econometrics explains the distribution of the sum of normally distributed variables (Dr. Jim McDonald calls it the useful theorem, so I don't actually know its real name). Applying the useful theorem to this problem reveals that

$$y_t \sim N\left(\mu_{z_t} + \mu_{v_t}, \Sigma_{z_t} + \Sigma_{v_t} - 2\text{cov}(z_t, v_t)\right)$$

In this case we can simplify the expression for the variance because we know z_t and v_t are independent, so $\text{cov}(z_t, v_t) = 0$. We are given that $\Sigma_{v_t} = 1 \forall t$, which is constant across time. Therefore, the distribution of y_t can be simplified as:

$$y_t \sim N(\mu_{z_t}, \Sigma_{z_t} + 1)$$

This identifies the matrix Ω as $\Sigma_z + 1$.

As in the model studied by Muth (described in section 2.8.1), the conditional expectation $E[y_t|y^{t-1}]$ can be given by equation 2.8.6b, which is

$$E[y_t|y^{t-1}] = GA^j \hat{x}_t$$

Making the necessary substitutions for this problem we have

$$E[y_t|y^{t-1}] = 0.9^j \hat{x}_t$$

$\square$

[a] The last two steps in the above calculations were a bit tricky. I used the assumption that the mean of z was time independent, and that may have not been permissible in this situation.