

ECE8823_hw2

February 7, 2019

1 7

1.1 a)

```
In [0]: import numpy as np
        np.set_printoptions(precision=2, suppress=True)

def make_sym_matrix(n,vals):
    m = np.zeros([n,n], dtype=np.double)
    xs,ys = np.triu_indices(n,k=0)
    m[xs,ys] = vals
    m[ys,xs] = vals
    return m

a12 = 2
a13 = -1
a34 = 0.5
a35 = -0.1

cov_list = []
cov_list.append(1) # cov(X1,X1)
cov_list.append(a12) # cov(X1, a12*X1+E2)
cov_list.append(a13) # cov(X1, a13*X1+E3)
cov_list.append(a13*a34) # cov(X1, a34*a13*X1+a34*E3+E4)
cov_list.append(a13*a35) # cov(X1, a35*a13*X1+a35*E3+E5)
cov_list.append(a12**2 + 1) #cov(a12*X1+E2, a12*X1+E2)
cov_list.append(a12*a13) #cov(a12*X1+E2, a13*X1+E3)
cov_list.append(a12*a13*a34) #cov(a12*X1+E2, a34*a13*X1+a34*E3+E4)
cov_list.append(a12*a13*a35) #cov(a12*X1+E2, a35*a13*X1+a35*E3+E5)
cov_list.append(a13**2 + 1) #cov(a13*X1+E3, a13*X1+E3)
cov_list.append((a13**2)*a34 + a34) #cov(a13*X1+E3, a34*a13*X1+a34*E3+E4)
cov_list.append((a13**2)*a35 + a35) #cov(a13*X1+E3, a35*a13*X1+a35*E3+E5)
cov_list.append((a13**2)*(a34**2) + a34**2 + 1) #cov(a34*a13*X1+a34*E3+E4, a34*a13*X1+a34*E3+E4)
cov_list.append((a13**2)*(a34*a35) + a34*a35) #cov(a34*a13*X1+a34*E3+E4, a35*a13*X1+a35*E3+E5)
cov_list.append((a13**2)*(a35**2) + a35**2 + 1) #cov(a35*a13*X1+a35*E3+E5, a35*a13*X1+a35*E3+E5)

R = make_sym_matrix(5,cov_list)
print(R)
```

```

S = np.linalg.inv(R)
print(S)

[[ 1.    2.   -1.   -0.5   0.1 ]
 [ 2.    5.   -2.   -1.    0.2 ]
 [-1.   -2.    2.    1.   -0.2 ]
 [-0.5  -1.    1.    1.5  -0.1 ]
 [ 0.1   0.2  -0.2  -0.1   1.02]]

[[ 6.   -2.    1.    0.    0. ]
 [-2.    1.   -0.   -0.   -0. ]
 [ 1.   -0.   1.26 -0.5   0.1 ]
 [ 0.    0.  -0.5   1.    0. ]
 [ 0.   -0.   0.1   0.    1. ]]

```

1.2 b)

```

In [0]: k = 1000
        a12 = 2
        a13 = -1
        a34 = 0.5
        a35 = -0.1
        x = np.zeros((k,5))
        R_hat = np.zeros((5,5))
        mse = lambda x,y: ((x-y)**2).mean()
        E = np.random.randn
        for kk in range(k):
            x[kk, 0] = E()
            x[kk, 1] = a12*x[kk, 0] + E()
            x[kk, 2] = a13*x[kk, 0] + E()
            x[kk, 3] = a34*x[kk, 2] + E()
            x[kk, 4] = a35*x[kk, 2] + E()
            _x = np.expand_dims(x[kk,:],1)
            R_hat += _x@_x.T/k
        print(R_hat)
        S_hat = np.linalg.inv(R_hat)
        print(S_hat)
        print(mse(R,R_hat))

[[ 1.    1.97 -0.99 -0.49  0.11]
 [ 1.97  4.88 -1.94 -0.95  0.3 ]
 [-0.99 -1.94  1.94  0.96 -0.21]
 [-0.49 -0.95  0.96  1.4  -0.15]
 [ 0.11  0.3  -0.21 -0.15  0.98]]

[[ 6.14 -2.05  1.11  0.    0.15]
 [-2.05  1.03 -0.03 -0.01 -0.08]
 [ 1.11 -0.03  1.34 -0.54  0.08]
 [ 0.   -0.01 -0.54  1.08  0.05]

```

```
[ 0.15 -0.08  0.08  0.05  1.05]]  
0.0028538716719830994
```

1.3 c)

```
In [0]: import cvxpy as cp
```

```
X = cp.Variable((5,5), PSD=True)  
objective = cp.Minimize(-cp.log_det(X)+cp.trace(X@R_hat))  
constraints = [X[S==0]==0]  
prob = cp.Problem(objective, constraints)  
result = prob.solve()  
S_cvx = X.value  
R_cvx = np.linalg.inv(S_cvx)  
print(R_cvx)  
print(S_cvx)  
print(mse(R,R_cvx))
```

```
[[ 1.    1.97 -0.99 -0.49  0.11]  
 [ 1.97  4.88 -1.96 -0.98  0.21]  
 [-0.99 -1.96  1.94  0.96 -0.21]  
 [-0.49 -0.98  0.96  1.4  -0.1 ]  
 [ 0.11  0.21 -0.21 -0.1  0.98]]  
[[ 6.02 -2.01  1.05 -0.   -0.  ]  
 [-2.01  1.02 -0.   -0.   0.  ]  
 [ 1.05 -0.   1.33 -0.54  0.11]  
 [-0.   -0.   -0.54  1.08 -0.  ]  
 [-0.   0.    0.11 -0.   1.04]]  
0.001521376622772955
```