cov-matrix-processing

July 1, 2022

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
cov_nercome = np.loadtxt("../output/Patchy_V6C_BOSS_DR12_NGC_z1_cov_nercome_250.
 ↪matrix")
cov_sample = np.loadtxt("../output/Patchy_V6C_BOSS_DR12_NGC_z1_cov_sample_250.
 ↪matrix")

# 'Real' covariance matrix before processing.
# The term 'real' here indicates that this is the matrix that we use to compare
# our estimates to.
cov_real_preproc = np.loadtxt("../data/
 ↪C_2048_BOSS_DR12_NGC_z1_V6C_1_1_1_1_1_10_200_200_prerecon.matrix")
```

[3]:
```
print(cov_nercome)
print(cov_sample)
```

```
[[ 2.33510456e+08  1.69725730e+07  6.42925467e+06 …  2.10843752e+04
  -3.84576887e+04 -7.64840880e+04]
 [ 1.69725730e+07  5.84577149e+07  7.44719805e+06 …  3.89195092e+04
   7.20969746e+04 -3.42312656e+04]
 [ 6.42925467e+06  7.44719805e+06  1.57188960e+07 … -1.24143002e+04
   6.01656089e+03 -8.09306563e+03]
 …
 [ 2.10843752e+04  3.89195092e+04 -1.24143002e+04 …  9.28236247e+03
   1.79869209e+03  1.08753651e+03]
 [-3.84576887e+04  7.20969746e+04  6.01656089e+03 …  1.79869209e+03
   8.29813556e+03  1.28750724e+03]
 [-7.64840880e+04 -3.42312656e+04 -8.09306563e+03 …  1.08753651e+03
   1.28750724e+03  8.54596602e+03]]
[[ 2.31827176e+08  1.74527581e+07  6.58064304e+06 …  2.17875839e+04
  -3.57859934e+04 -7.49325641e+04]
 [ 1.74527581e+07  5.52690451e+07  7.31856179e+06 …  3.84148375e+04
   6.71573030e+04 -3.27511889e+04]
 [ 6.58064304e+06  7.31856179e+06  1.42549217e+07 … -1.03600748e+04
   5.72259895e+03 -8.07066239e+03]
 …
```

```
 [ 2.17875839e+04  3.84148375e+04 -1.03600748e+04 …  4.38288013e+03
   1.18191532e+03  7.60942314e+02]
 [-3.57859934e+04  6.71573030e+04  5.72259895e+03 …  1.18191532e+03
   3.75088648e+03  8.88091933e+02]
 [-7.49325641e+04 -3.27511889e+04 -8.07066239e+03 …  7.60942314e+02
   8.88091933e+02  3.83115144e+03]]
```

[4]:
```
indices = np.concatenate((np.arange(40), np.arange(40)+80, np.arange(40)+160))
cov_real = (cov_real_preproc[indices, :])[:, indices]
```

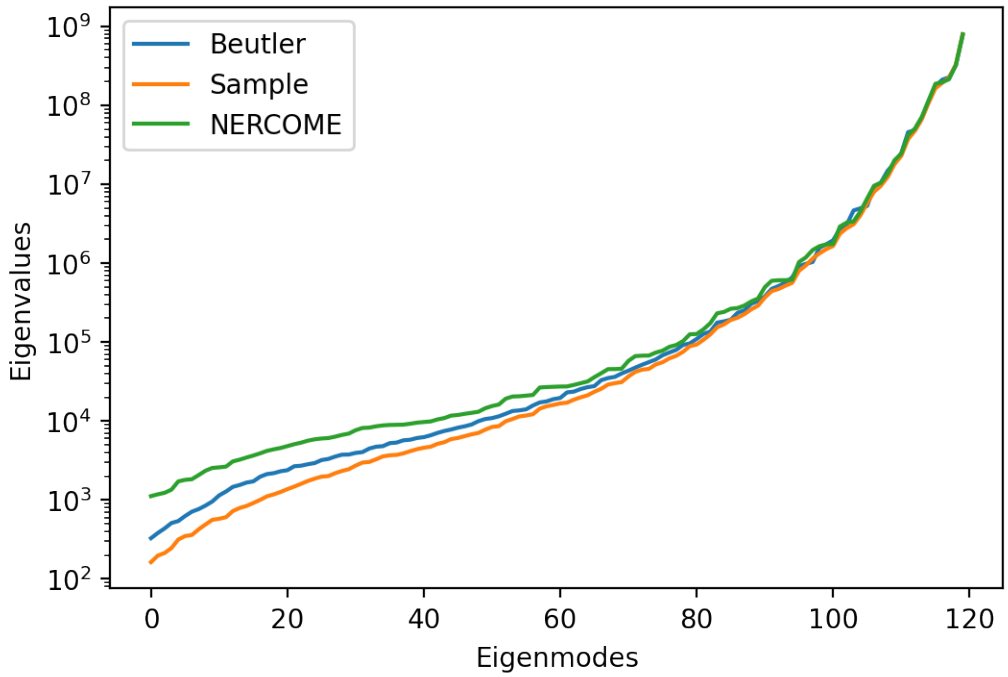
[5]:
```
print(cov_real)
```

```
[[ 2.39531896e+08  1.91051563e+07  1.44038639e+05 …  9.68083967e+03
  -1.21684291e+04  2.21567909e+04]
 [ 1.91051563e+07  5.15256710e+07  4.78376836e+06 …  9.13608077e+03
   9.26311893e+03 -1.41639754e+04]
 [ 1.44038639e+05  4.78376836e+06  1.67772970e+07 … -5.49127765e+02
  -2.28200815e+03  6.76773412e+03]
 …
 [ 9.68083967e+03  9.13608077e+03 -5.49127765e+02 …  4.22979902e+03
   1.28931823e+03  3.78440184e+02]
 [-1.21684291e+04  9.26311893e+03 -2.28200815e+03 …  1.28931823e+03
   3.88691287e+03  1.00320951e+03]
 [ 2.21567909e+04 -1.41639754e+04  6.76773412e+03 …  3.78440184e+02
   1.00320951e+03  3.62984716e+03]]
```

[6]:
```
evals_nercome = np.linalg.eigvalsh(cov_nercome)
evals_sample = np.linalg.eigvalsh(cov_sample)
evals_real = np.linalg.eigvalsh(cov_real)
```

[7]:
```
plt.figure(dpi=200)
plt.plot(evals_real, label="Beutler")
plt.plot(evals_sample, label="Sample")
plt.plot(evals_nercome, label="NERCOME")
plt.yscale("log")
plt.legend()
plt.xlabel("Eigenmodes")
plt.ylabel("Eigenvalues")
```

[7]:
```
Text(0, 0.5, 'Eigenvalues')
```



[8]:
```
nercome_rel_errors = (cov_nercome-cov_real)/np.abs(cov_real)
print(nercome_rel_errors)
```

```
[[-2.51383678e-02 -1.11623439e-01  4.36356251e+01 …  1.17794901e+00
  -2.16044810e+00 -4.45194791e+00]
 [-1.11623439e-01  1.34535733e-01  5.56763934e-01 …  3.25997867e+00
   6.78322886e+00 -1.41678376e+00]
 [ 4.36356251e+01  5.56763934e-01 -6.30853102e-02 … -2.16073075e+01
   3.63652034e+00 -2.19583091e+00]
 …
 [ 1.17794901e+00  3.25997867e+00 -2.16073075e+01 …  1.19451620e+00
   3.95072252e-01  1.87373423e+00]
 [-2.16044810e+00  6.78322886e+00  3.63652034e+00 …  3.95072252e-01
   1.13489107e+00  2.83388192e-01]
 [-4.45194791e+00 -1.41678376e+00 -2.19583091e+00 …  1.87373423e+00
   2.83388192e-01  1.35435974e+00]]
```

[9]:
```
nercome_pos_count = 0
nercome_neg_count = 0
for row in nercome_rel_errors:
    for error in row:
        if error >= 0:
            nercome_pos_count += 1
        elif error < 0:
            nercome_neg_count += 1
```

```
print(f"NERCOME overestimated {nercome_pos_count} elements")
print(f"NERCOME underestimated {nercome_neg_count} elements")

index_max_nercome = np.unravel_index(np.abs(nercome_rel_errors).argmax(),␣
 ↪nercome_rel_errors.shape)

print(f"Maximum relative error is {nercome_rel_errors[index_max_nercome]}")
print(f"Maximum relative error index is {index_max_nercome}")
print(f"NERCOME: {cov_nercome[index_max_nercome]}, real:␣
 ↪{cov_real[index_max_nercome]}")
```

```
NERCOME overestimated 8399 elements
NERCOME underestimated 6001 elements
Maximum relative error is -26736.59033744984
Maximum relative error index is (25, 54)
NERCOME: -983.2713417795338, real: -0.03677486749440995
```

[10]:
```
sample_rel_errors = (cov_sample-cov_real)/np.abs(cov_real)
print(sample_rel_errors)
```

```
[[-3.21657395e-02 -8.64896433e-02  4.46866512e+01 …  1.25058824e+00
  -1.94088851e+00 -4.38192315e+00]
 [-8.64896433e-02  7.26506606e-02  5.29873782e-01 …  3.20473925e+00
   6.24996661e+00 -1.31228790e+00]
 [ 4.46866512e+01  5.29873782e-01 -1.50344556e-01 … -1.78664195e+01
   3.50770312e+00 -2.19252060e+00]
 …
 [ 1.25058824e+00  3.20473925e+00 -1.78664195e+01 …  3.61911076e-02
  -8.33020964e-02  1.01073339e+00]
 [-1.94088851e+00  6.24996661e+00  3.50770312e+00 … -8.33020964e-02
  -3.49959949e-02 -1.14749292e-01]
 [-4.38192315e+00 -1.31228790e+00 -2.19252060e+00 …  1.01073339e+00
  -1.14749292e-01  5.54580597e-02]]
```

[11]:
```
sample_pos_count = 0
sample_neg_count = 0
for row in sample_rel_errors:
    for error in row:
        if error >= 0:
            sample_pos_count += 1
        elif error < 0:
            sample_neg_count += 1

print(f"Sample overestimated {sample_pos_count} elements")
print(f"Sample underestimated {sample_neg_count} elements")
```

```
index_max_sample = np.unravel_index(np.abs(sample_rel_errors).argmax(),␣
 ↪sample_rel_errors.shape)

print(f"Maximum relative error is {sample_rel_errors[index_max_sample]}")
print(f"Maximum relative error index is {index_max_sample}")
print(f"Sample: {cov_sample[index_max_sample]}, real:␣
 ↪{cov_real[index_max_sample]}")
```

```
Sample overestimated 7789 elements
Sample underestimated 6611 elements
Maximum relative error is -17676.38763772095
Maximum relative error index is (54, 25)
Sample: -650.0835880245085, real: -0.03677486749440995
```

[12]:
```
MSE_NERCOME = np.trace((cov_nercome-cov_real)@(cov_nercome-cov_real).T)
MSE_sample = np.trace((cov_sample-cov_real)@(cov_sample-cov_real).T)
print(f"MSE NERCOME: {MSE_NERCOME}")
print(f"MSE sample: {MSE_sample}")
```

```
MSE NERCOME: 1.648741445040605e+16
MSE sample: 1.7807956323650856e+16
```

[ ]:
```

```