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_2048.matrix")
cov_sample = np.loadtxt("../output/Patchy_V6C_BOSS_DR12_NGC_z1_cov_sample_2048.
 ↪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)
```

```
[[ 2.37323130e+08  1.86024630e+07  9.93829745e+04 …  9.20947171e+03
  -1.24516461e+04  2.24203892e+04]
 [ 1.86024630e+07  5.16992164e+07  4.74738440e+06 …  9.42837843e+03
   9.52712278e+03 -1.41014722e+04]
 [ 9.93829745e+04  4.74738440e+06  1.69697364e+07 … -4.73772115e+02
  -2.34027554e+03  6.80012721e+03]
 …
 [ 9.20947171e+03  9.42837843e+03 -4.73772115e+02 …  4.70457537e+03
   1.36829796e+03  4.09849815e+02]
 [-1.24516461e+04  9.52712278e+03 -2.34027554e+03 …  1.36829796e+03
   4.35627101e+03  1.08072377e+03]
 [ 2.24203892e+04 -1.41014722e+04  6.80012721e+03 …  4.09849815e+02
   1.08072377e+03  4.07136743e+03]]
[[ 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]]
```

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

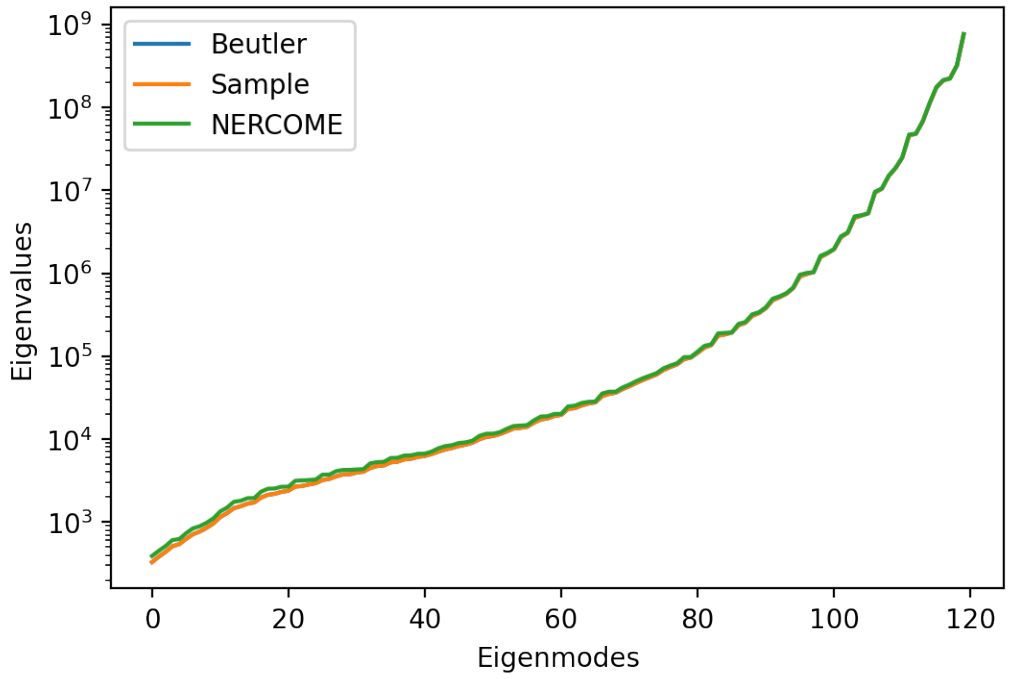
[5]:
```
print(cov_real)
```

```
[[ 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)
```

```
[[-0.00922118 -0.02631192 -0.31002559 … -0.04869081 -0.02327474
   0.01189695]
 [-0.02631192  0.00336813 -0.00760571 …  0.03199377  0.02850054
   0.00441283]
 [-0.31002559 -0.00760571  0.01147023 …  0.1372279  -0.02553338
   0.0047864 ]
 …
 [-0.04869081  0.03199377  0.1372279  …  0.1122456   0.06125698
   0.08299761]
 [-0.02327474  0.02850054 -0.02553338 …  0.06125698  0.12075345
   0.07726627]
 [ 0.01189695  0.00441283  0.0047864  …  0.08299761  0.07726627
   0.12163605]]
```

[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 8846 elements
NERCOME underestimated 5554 elements
Maximum relative error is -962.7092816081185
Maximum relative error index is (54, 25)
NERCOME: -35.440281134271565, real: -0.03677486749440995
```

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

```
[[ 1.24419014e-16  1.94988737e-16  1.01027859e-15 … -1.87895830e-16
   0.00000000e+00 -3.28384992e-16]
 [ 3.89977475e-16 -1.44599390e-16  1.94683878e-16 …  1.99099532e-16
   5.89107001e-16 -2.56847298e-16]
 [ 7.47606155e-15  1.94683878e-16  2.22043533e-16 …  4.14063338e-16
  -2.39130093e-15 -1.34386884e-16]
 …
 [-5.63687490e-16 -1.99099532e-16  8.28126676e-16 … -2.15020784e-16
   1.76351866e-16 -4.50613502e-16]
 [ 2.98968648e-16  9.81845001e-16 -1.59420062e-15 …  1.76351866e-16
  -1.16994480e-16  1.13323126e-16]
 [-1.64192496e-16  2.56847298e-16 -1.34386884e-16 … -1.50204501e-16
   1.13323126e-16  0.00000000e+00]]
```

[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 8087 elements
Sample underestimated 6313 elements
Maximum relative error is 7.58384684111064e-12
Maximum relative error index is (25, 54)
Sample: -0.03677486749413106, 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: 52247720917254.195
MSE sample: 2.2537432416564147e-14
```

[ ]:
```

```