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_1000.matrix")
cov_sample = np.loadtxt("../output/Patchy_V6C_BOSS_DR12_NGC_z1_cov_sample_1000.
 ↪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)
```

```
[[ 2.34958981e+08  1.65151221e+07  1.21395750e+06 …  2.51592737e+04
   4.10909186e+04  6.24529073e+03]
 [ 1.65151221e+07  5.28226375e+07  5.99617779e+06 …  6.44345911e+03
  -2.88347924e+03 -1.73785088e+04]
 [ 1.21395750e+06  5.99617779e+06  1.71920515e+07 …  4.16298856e+03
  -2.13849840e+03  1.45449988e+04]
 …
 [ 2.51592737e+04  6.44345911e+03  4.16298856e+03 …  5.30755404e+03
   1.29987248e+03  3.34593066e+02]
 [ 4.10909186e+04 -2.88347924e+03 -2.13849840e+03 …  1.29987248e+03
   4.75895350e+03  1.31144297e+03]
 [ 6.24529073e+03 -1.73785088e+04  1.45449988e+04 …  3.34593066e+02
   1.31144297e+03  4.61589423e+03]]
[[ 2.34599171e+08  1.65976463e+07  1.23274680e+06 …  2.61447159e+04
   4.13009204e+04  6.84916373e+03]
 [ 1.65976463e+07  5.26292409e+07  6.03247639e+06 …  5.55896408e+03
  -3.71761364e+03 -1.82098635e+04]
 [ 1.23274680e+06  6.03247639e+06  1.68205859e+07 …  4.13186738e+03
  -2.12313103e+03  1.41682731e+04]
 …
```

```
 [ 2.61447159e+04  5.55896408e+03  4.13186738e+03 …  4.29010757e+03
   1.13795758e+03  3.41027694e+02]
 [ 4.13009204e+04 -3.71761364e+03 -2.12313103e+03 …  1.13795758e+03
   3.82604763e+03  1.14398202e+03]
 [ 6.84916373e+03 -1.82098635e+04  1.41682731e+04 …  3.41027694e+02
   1.14398202e+03  3.67578056e+03]]
```

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

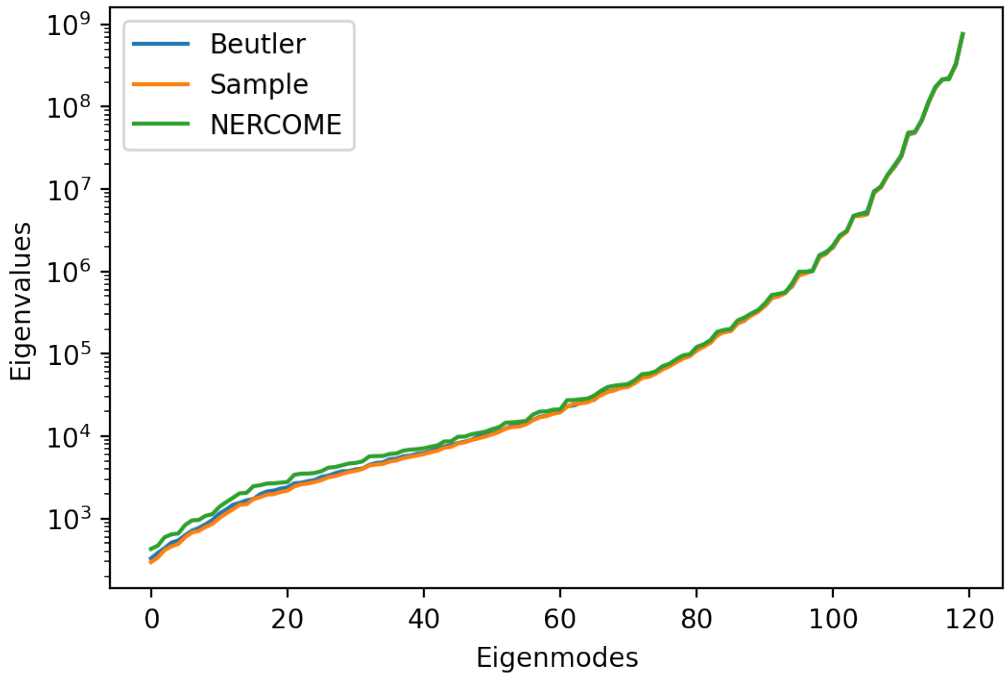
[5]:
```
print(cov_real)
```

```
[[ 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)
```

```
[[-1.90910491e-02 -1.35567286e-01  7.42799901e+00 …  1.59887309e+00
   4.37684661e+00 -7.18131983e-01]
 [-1.35567286e-01  2.51712689e-02  2.53442336e-01 … -2.94723934e-01
  -1.31128600e+00 -2.26951355e-01]
 [ 7.42799901e+00  2.53442336e-01  2.47211751e-02 …  8.58109283e+00
   6.28874813e-02  1.14916817e+00]
 …
 [ 1.59887309e+00 -2.94723934e-01  8.58109283e+00 …  2.54800528e-01
   8.18591511e-03 -1.15862743e-01]
 [ 4.37684661e+00 -1.31128600e+00  6.28874813e-02 …  8.18591511e-03
   2.24353017e-01  3.07247340e-01]
 [-7.18131983e-01 -2.26951355e-01  1.14916817e+00 … -1.15862743e-01
   3.07247340e-01  2.71649747e-01]]
```

[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 7278 elements
NERCOME underestimated 7122 elements
Maximum relative error is 17396.64644677944
Maximum relative error index is (25, 54)
NERCOME: 639.7225930599172, real: -0.03677486749440995
```

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

```
[[-2.05931875e-02 -1.31247814e-01  7.55844524e+00 …  1.70066614e+00
   4.39410453e+00 -6.90877449e-01]
 [-1.31247814e-01  2.14178647e-02  2.61030204e-01 … -3.91537332e-01
  -1.40133498e+00 -2.85646369e-01]
 [ 7.55844524e+00  2.61030204e-01  2.58021022e-03 …  8.52441899e+00
   6.96216282e-02  1.09350321e+00]
 …
 [ 1.70066614e+00 -3.91537332e-01  8.52441899e+00 …  1.42580173e-02
  -1.17395880e-01 -9.88597180e-02]
 [ 4.39410453e+00 -1.40133498e+00  6.96216282e-02 … -1.17395880e-01
  -1.56590170e-02  1.40322142e-01]
 [-6.90877449e-01 -2.85646369e-01  1.09350321e+00 … -9.88597180e-02
   1.40322142e-01  1.26543600e-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 6677 elements
Sample underestimated 7723 elements
Maximum relative error is 17205.47380302353
Maximum relative error index is (25, 54)
Sample: 632.6922444172377, 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: 1354396742410341.8
MSE sample: 1346439911240411.2
```

[ ]:
```

```