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_200.
 ↪matrix")
cov_sample = np.loadtxt("../output/Patchy_V6C_BOSS_DR12_NGC_z1_cov_sample_200.
 ↪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)
```

```
[[ 2.28506959e+08  6.42655535e+06 -1.06423240e+06 … -4.30511096e+03
  -6.29546771e+04 -3.67480992e+04]
 [ 6.42655535e+06  5.72291300e+07  1.89950886e+06 …  2.83303162e+04
   7.40813638e+04 -1.25777799e+04]
 [-1.06423240e+06  1.89950886e+06  2.11413010e+07 … -5.56466397e+03
  -2.42813916e+04 -2.76331372e+04]
 …
 [-4.30511096e+03  2.83303162e+04 -5.56466397e+03 …  1.13682080e+04
   2.89956568e+03 -6.22692178e+01]
 [-6.29546771e+04  7.40813638e+04 -2.42813916e+04 …  2.89956568e+03
   1.06798482e+04  1.39264414e+03]
 [-3.67480992e+04 -1.25777799e+04 -2.76331372e+04 … -6.22692178e+01
   1.39264414e+03  9.25917354e+03]]
[[ 2.19870449e+08  6.58146060e+06 -8.49756132e+05 … -6.25446109e+03
  -5.98302303e+04 -3.80366068e+04]
 [ 6.58146060e+06  5.14587823e+07  2.31434979e+06 …  2.35449432e+04
   6.52292161e+04 -1.18367386e+04]
 [-8.49756132e+05  2.31434979e+06  1.78541839e+07 … -4.12673501e+03
  -1.98163085e+04 -2.45021651e+04]
 …
```

```
[-6.25446109e+03  2.35449432e+04 -4.12673501e+03 …  4.11567779e+03
  1.53786167e+03  2.02838640e-01]
[-5.98302303e+04  6.52292161e+04 -1.98163085e+04 …  1.53786167e+03
  3.71231389e+03  6.81792792e+02]
[-3.80366068e+04 -1.18367386e+04 -2.45021651e+04 …  2.02838640e-01
  6.81792792e+02  2.80471896e+03]]
```

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

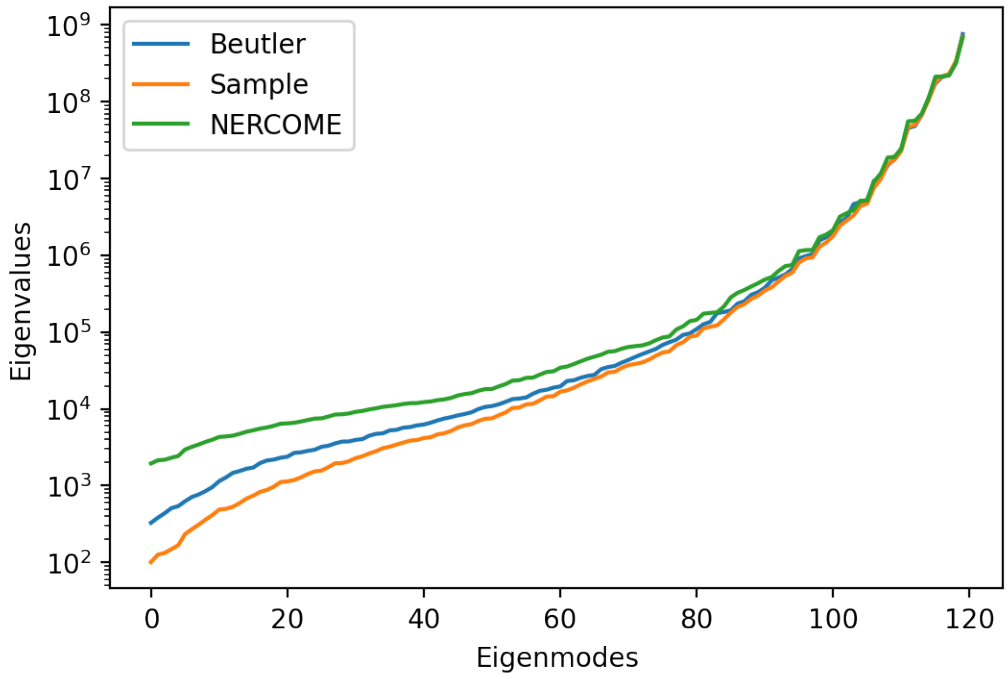
[5]:
```
print(cov_real)
```

```
[[ 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)
```

```
[[-0.04602701 -0.66362194 -8.38852026 … -1.44470429 -4.17360757
  -2.65854791]
 [-0.66362194  0.1106916  -0.60292625 …  2.10092663  6.99745359
   0.11198802]
 [-8.38852026 -0.60292625  0.26011365 … -9.13364161 -9.6403615
  -5.08307074]
 …
 [-1.44470429  2.10092663 -9.13364161 …  1.68764732  1.24891389
  -1.16454177]
 [-4.17360757  6.99745359 -9.6403615  …  1.24891389  1.74764282
   0.38818873]
 [-2.65854791  0.11198802 -5.08307074 … -1.16454177  0.38818873
   1.55084391]]
```

[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 7228 elements
NERCOME underestimated 7172 elements
Maximum relative error is -53538.754166466664
Maximum relative error index is (25, 54)
NERCOME: -1968.9173651550948, real: -0.03677486749440995
```

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

```
[[-8.20827925e-02 -6.55513909e-01 -6.89950127e+00 … -1.64606597e+00
  -3.91684092e+00 -2.71670198e+00]
 [-6.55513909e-01 -1.29816282e-03 -5.16207807e-01 …  1.57713825e+00
   6.04181999e+00  1.64306749e-01]
 [-6.89950127e+00 -5.16207807e-01  6.41871519e-02 … -6.51507258e+00
  -7.68371504e+00 -4.62043849e+00]
 …
 [-1.64606597e+00  1.57713825e+00 -6.51507258e+00 … -2.69802957e-02
   1.92771215e-01 -9.99464014e-01]
 [-3.91684092e+00  6.04181999e+00 -7.68371504e+00 …  1.92771215e-01
  -4.49197029e-02 -3.20388431e-01]
 [-2.71670198e+00  1.64306749e-01 -4.62043849e+00 … -9.99464014e-01
  -3.20388431e-01 -2.27317615e-01]]
```

[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 6352 elements
Sample underestimated 8048 elements
Maximum relative error is 47402.094043586454
Maximum relative error index is (80, 94)
Sample: 631168.4055908481, real: -13.315481811674502
```

[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: 2.814881523245139e+16
MSE sample: 2.6448169256504176e+16
```

[ ]:
```

```