1.3

December 3, 2021

[21]:
```
import numpy as np
from matplotlib import pyplot as plt
from skimage import io
import warnings
warnings.filterwarnings("ignore")
from astroML.datasets import sdss_corrected_spectra
```

1 1.3.1:

- The projections of ISOMAP did not change with the change in number of components.
- The projections of UMAP changed as the number of components changed. The appear to lose the u-shape and became thinner and condensed as the number of components increased.
- The projections of Spectral Embedding did not change with the change in number of components.
- The LLE projections changed as the number of components changed. The change is in these projections were as follows: the projections first formed a straight line with a negative gradient, then they proceeded to form an L-shape like shape and became a single dot.

2 1.3.2:

The ISOMAP and Spectral Embedding projections were more stable as the number of components increased.

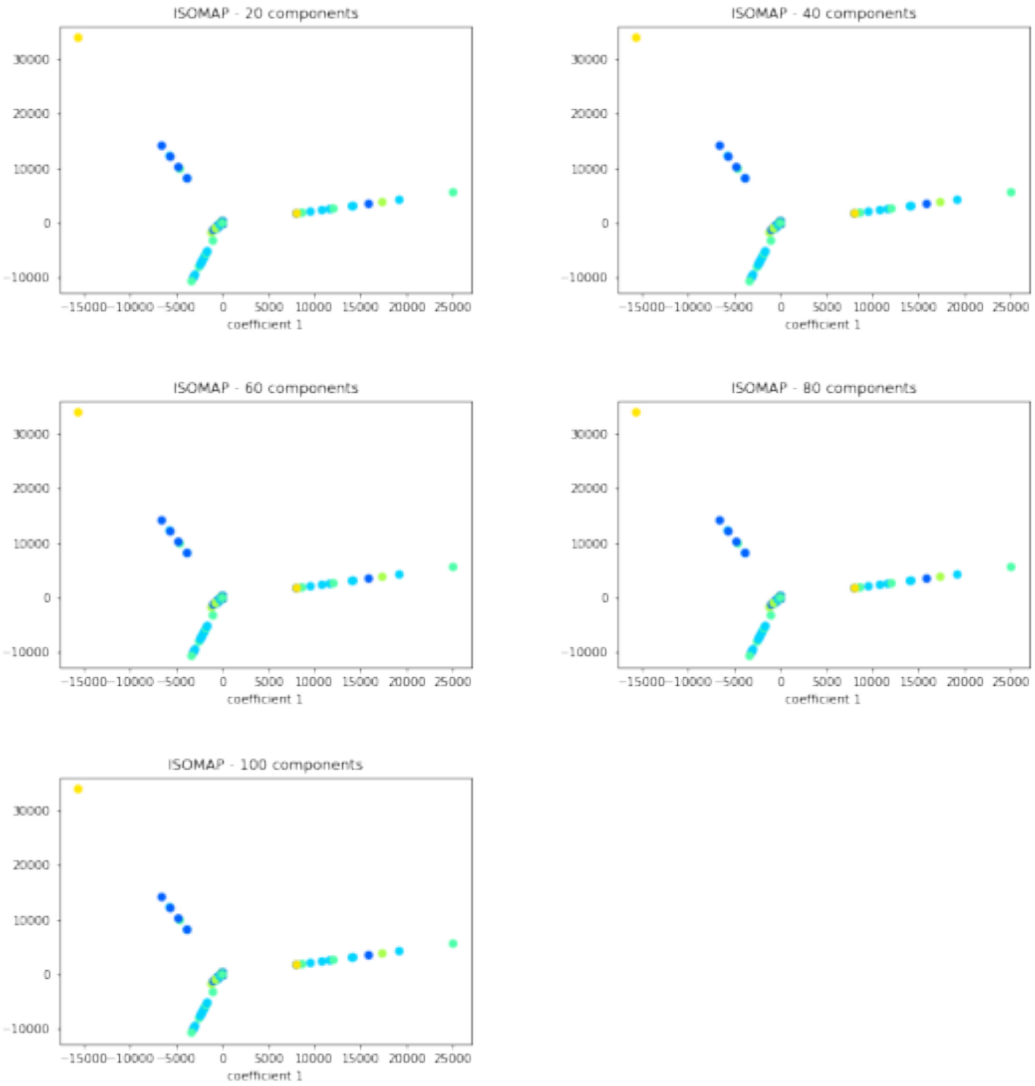
3 1.3.3

3.1 Isomap:

The following plots show the different Isomap projections produced for different component sizes specifically, 20, 40, 60, 80, 100 components. The number of neighbours were kept constant at 100.

[2]:
```
fig,axes = plt.subplots(nrows=3, ncols=2, figsize=(8,8), sharex=True,␣
↪sharey=True)
ax=axes.ravel()


ax[0].imshow(io.imread("1.3 plots/ISOMAP - 20 components.png"), cmap=plt.cm.
↪gray)
```

```
ax[1].imshow(io.imread("1.3 plots/ISOMAP - 40 components.png"), cmap=plt.cm.
 ↪gray)


ax[2].imshow(io.imread("1.3 plots/ISOMAP - 60 components.png"), cmap=plt.cm.
 ↪gray)

ax[3].imshow(io.imread("1.3 plots/ISOMAP - 80 components.png"), cmap=plt.cm.
 ↪gray)

ax[4].imshow(io.imread("1.3 plots/ISOMAP - 100 components.png"), cmap=plt.cm.
 ↪gray)

for a in ax:
    a.axis('off')

fig.tight_layout()
plt.show()
```

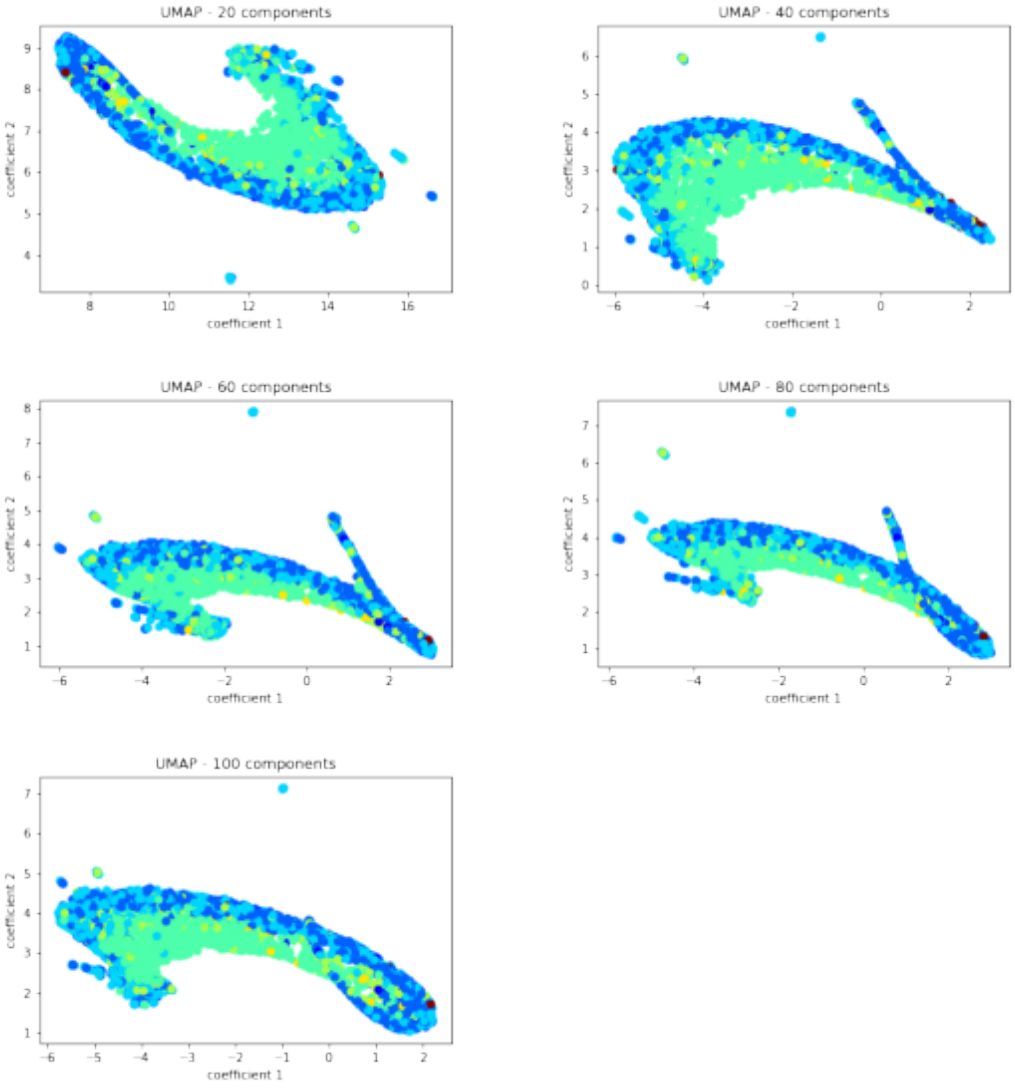


3.2 UMAPS:

The following plots show the different UMAP projections produced for different component sizes specifically, 20, 40, 60, 80, 100 components. The number of neighbours were kept constant at 100.

[3]:
```
fig,axes = plt.subplots(nrows=3, ncols=2, figsize=(8,8), sharex=True,␣
↪sharey=True)
ax=axes.ravel()


ax[0].imshow(io.imread("1.3 plots/UMAP - 20 components.png"), cmap=plt.cm.gray)

ax[1].imshow(io.imread("1.3 plots/UMAP - 40 components.png"), cmap=plt.cm.gray)
```

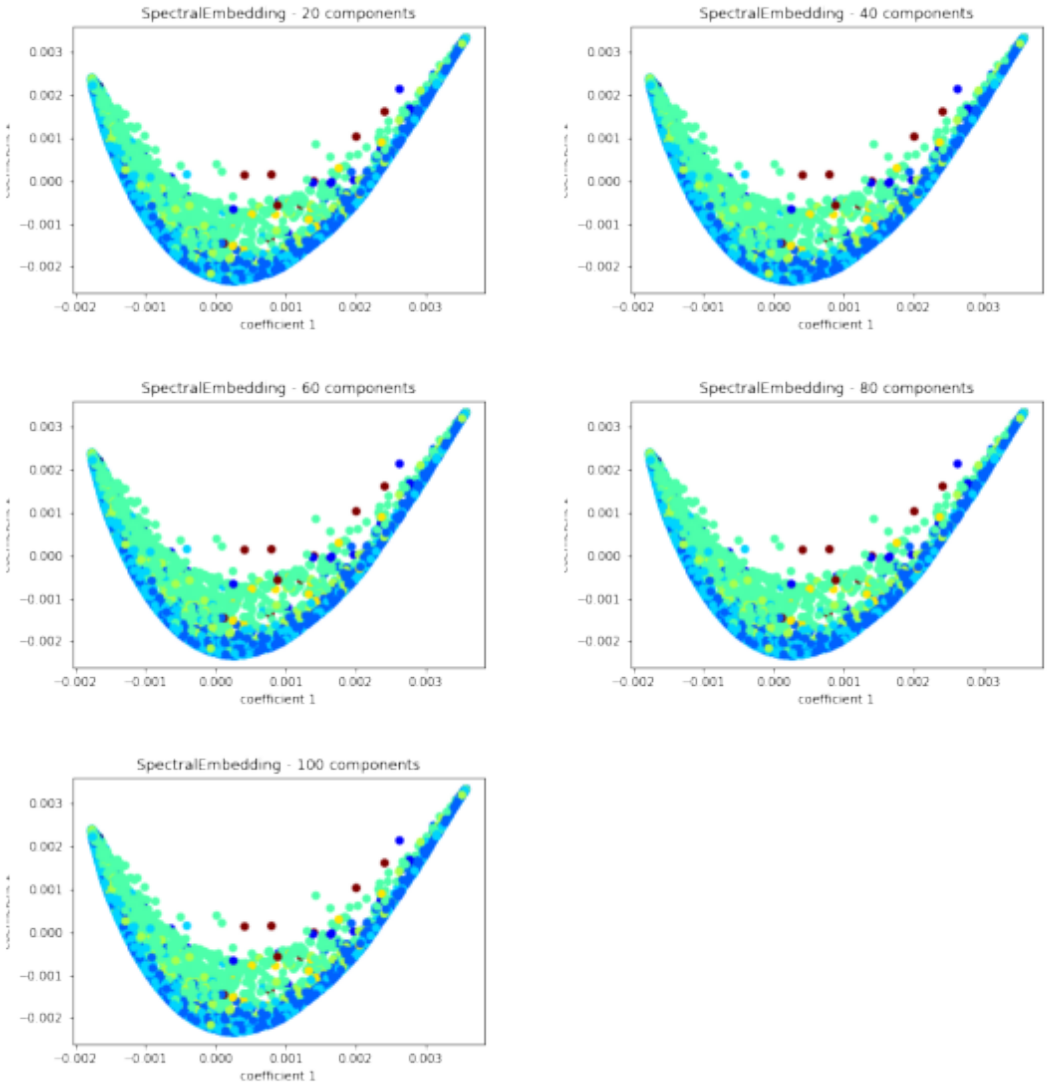
```
ax[2].imshow(io.imread("1.3 plots/UMAP - 60 components.png"), cmap=plt.cm.gray)

ax[3].imshow(io.imread("1.3 plots/UMAP - 80 components.png"), cmap=plt.cm.gray)

ax[4].imshow(io.imread("1.3 plots/UMAP - 100 components.png"), cmap=plt.cm.gray)

for a in ax:
    a.axis('off')

fig.tight_layout()
plt.show()
```



3.3 Spectral Embedding:

The following plots show the different Spectral Embedding projections produced for different component sizes specifically, 20, 40, 60, 80, 100 components. The number of neighbours were kept constant at 100.

[4]:
```
fig,axes = plt.subplots(nrows=3, ncols=2, figsize=(8,8), sharex=True,␣
 ↪sharey=True)
ax=axes.ravel()


ax[0].imshow(io.imread("1.3 plots/SpectralEmbedding - 20 components.png"),␣
 ↪cmap=plt.cm.gray)

ax[1].imshow(io.imread("1.3 plots/SpectralEmbedding - 40 components.png"),␣
 ↪cmap=plt.cm.gray)

ax[2].imshow(io.imread("1.3 plots/SpectralEmbedding - 60 components.png"),␣
 ↪cmap=plt.cm.gray)

ax[3].imshow(io.imread("1.3 plots/SpectralEmbedding - 80 components.png"),␣
 ↪cmap=plt.cm.gray)

ax[4].imshow(io.imread("1.3 plots/SpectralEmbedding - 100 components.png"),␣
 ↪cmap=plt.cm.gray)

for a in ax:
    a.axis('off')

fig.tight_layout()
plt.show()
```

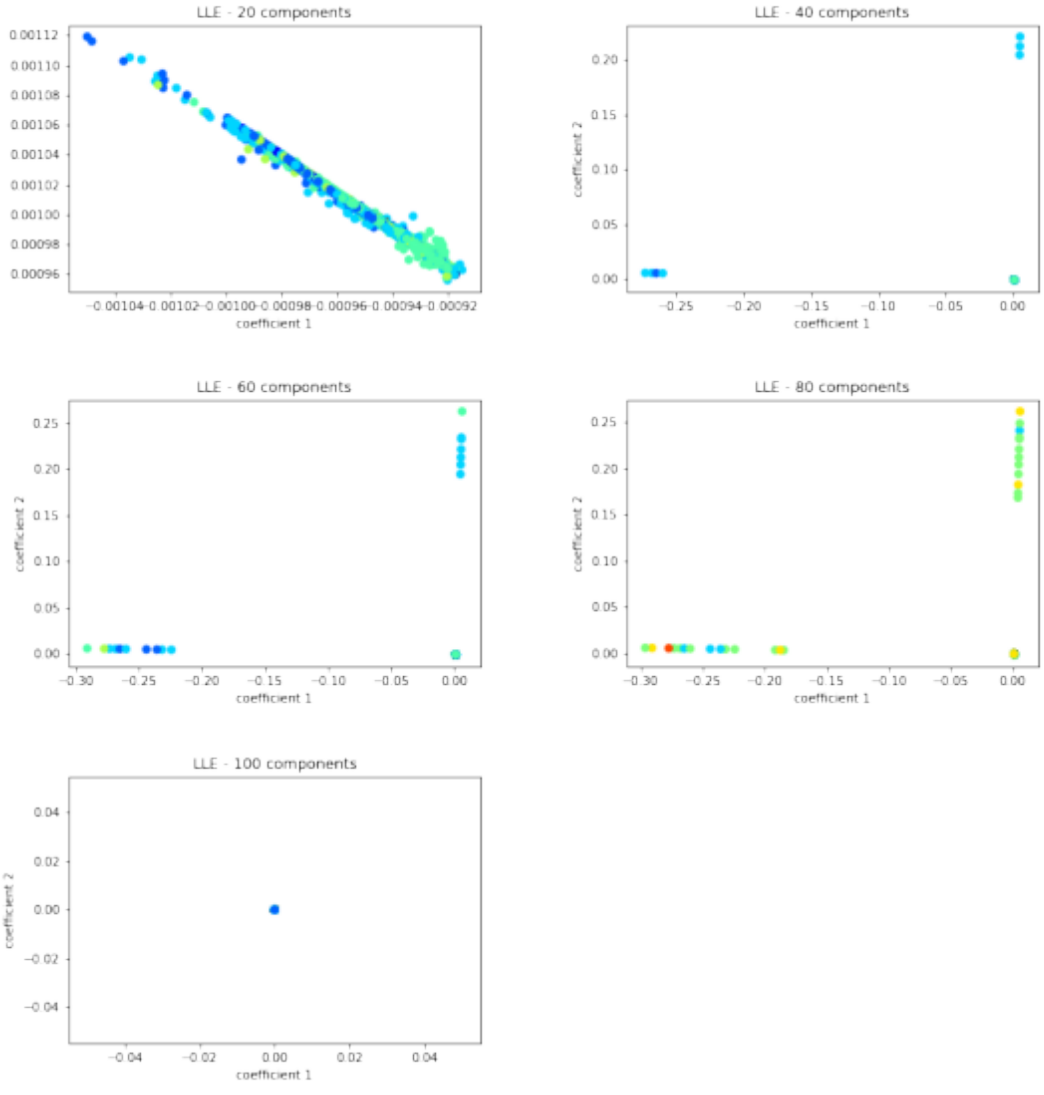


3.4 LLE:

The following plots show the different LLE projections produced for different component sizes specifically, 20, 40, 60, 80, 100 components. The number of neighbours were kept constant at 100.

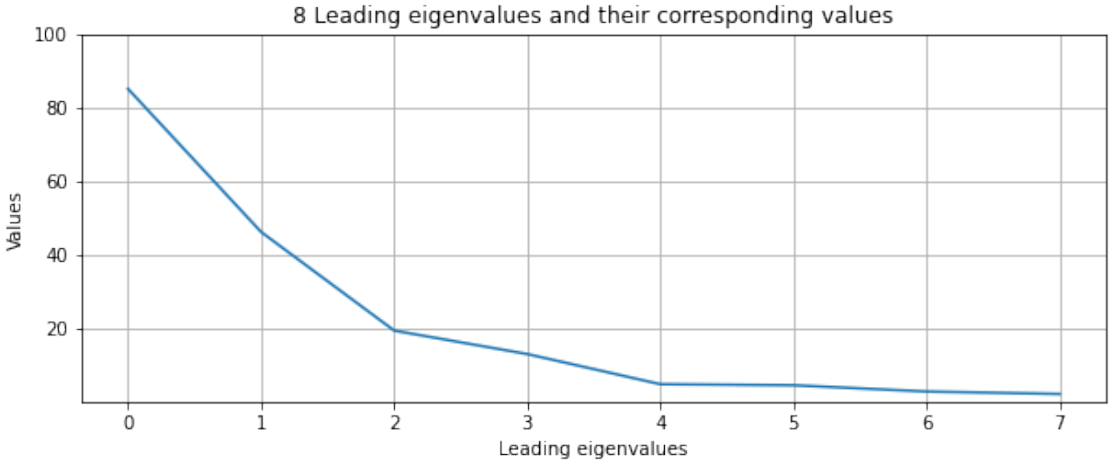
[5]:
```
fig,axes = plt.subplots(nrows=3, ncols=2, figsize=(8,8), sharex=True,␣
↪sharey=True)
ax=axes.ravel()


ax[0].imshow(io.imread("1.3 plots/LLE - 20 components.png"), cmap=plt.cm.gray)

ax[1].imshow(io.imread("1.3 plots/LLE - 40 components.png"), cmap=plt.cm.gray)
```

```
ax[2].imshow(io.imread("1.3 plots/LLE - 60 components.png"), cmap=plt.cm.gray)

ax[3].imshow(io.imread("1.3 plots/LLE - 80 components.png"), cmap=plt.cm.gray)

ax[4].imshow(io.imread("1.3 plots/LLE - 100 components.png"), cmap=plt.cm.gray)

for a in ax:
    a.axis('off')

fig.tight_layout()
plt.show()
```



4 1.3.4

[22]:
```
data = sdss_corrected_spectra.fetch_sdss_corrected_spectra()
evals = data['evals'] ** 2

leading_evals = evals[evals>= 1]
fig = plt.figure(figsize=(10, 7.5))
fig.subplots_adjust(hspace=0.05, bottom=0.12)

ax = fig.add_subplot(211)
ax.grid()
ax.plot(leading_evals)
ax.set_ylabel('Values')
ax.set_ylim(5E-4, 100)
ax.set_xlabel("Leading eigenvalues")
plt.title("8 Leading eigenvalues and their corresponding values")
plt.savefig("Leading eigenvalues")
plt.show()
```



4.1 Comments:

It is the same.

[ ]:
```

```