

PCA_of_Xe_arc

November 25, 2019

1 Import necessary packages

```
In [1]: import os
import math
import numpy as np
from lsst.utils import getPackageDir
import lsst.daf.persistence as dafPersist
from pfs.drp.stella.utils import readLineListFile
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
```

2 Read saved line information

I have run `pfs.drp.stealla.findLines.FindLinesTask` on `pfsArm` files of Xe arc taken on May 2019 at LAM. Reading `pfsArm` files and conversion to `pfs.drp.stella.SpectrumSet` takes time. So I have saved `visit`, `fiberId`, `center_wavelength` and `flux` for the arc lines detected by `pfs.drp.stealla.findLines.FindLinesTask`.

```
In [2]: visit, fiberId, center_wl, flux = np.loadtxt('XeVector.dat', usecols=(0,1,2,3), unpack=True)
visit = list(map(int, visit))
fiberId = list(map(int, fiberId))

nVisit = len(set(visit))
nFiber = len(set(fiberId))
print('visit (%d)' % (nVisit), ':', sorted(set(visit)))
print('fiberId (%d)' % (nFiber), ':', sorted(set(fiberId)))
```

```
visit (18) : [17533, 17534, 17535, 17536, 17537, 17538, 17641, 17642, 17643, 17644, 17645, 17646, 17647, 17648, 17649, 17650, 17651, 17652]
fiberId (10) : [2, 63, 192, 255, 339, 401, 464, 525, 587, 650]
```

```
In [3]: centerWL = dict()
Fluxes = dict()

for v, fid, wl, fl in zip(visit, fiberId, center_wl, flux):
    if not (v, fid) in centerWL.keys():
        centerWL[(v, fid)] = list()
```

```

        Fluxes[(v, fid)] = list()

        centerWL[(v, fid)].append(wl)
        Fluxes[(v, fid)].append(fl)

    for key in centerWL.keys():
        centerWL[key] = np.array(centerWL[key])
        Fluxes[key] = np.array(Fluxes[key])

```

3 Read arc line list

```

In [4]: lineListFilename = os.path.join(getPackageDir("obs_pfs"), "pfs", "lineLists", "ArCd
        arcLines = readLineListFile(lineListFilename, "Xe")

```

4 Select arc lines which are common in all visits and fiberIds

```

In [5]: detectedWL = dict()
        for key in centerWL.keys():
            for wl in centerWL[key]:
                keep = False
                for arc in arcLines:
                    if math.fabs(wl - arc.wavelength) < 0.2:
                        keep = True
                        break

                if not keep:
                    continue

                if not arc.wavelength in detectedWL.keys():
                    detectedWL[arc.wavelength] = 0
                    detectedWL[arc.wavelength] += 1
        commonWL = sorted([wl for wl in detectedWL.keys() if detectedWL[wl] == nVisit * nFiber])
        commonWL = commonWL[0:41] + commonWL[42:] # Remove a line at 895.225. It seems to
        print('# of common wanvavelength : %d' % (len(commonWL)))

```

```

# of common wanvavelength : 52

```

5 Select fluxes corresponding to common wavelength

```

In [6]: commonFluxes = dict()
        for key in centerWL.keys():
            for cWL in commonWL:
                dwl = 100
                for wl, fl in zip(centerWL[key], Fluxes[key]):
                    if math.fabs(wl - cWL) < dwl:

```

```

        dwl = math.fabs(wl - cWL)
        fl_a = fl

        if not key in commonFluxes.keys():
            commonFluxes[key] = list()
            commonFluxes[key].append(fl_a)

for key in commonFluxes.keys():
    commonFluxes[key] = np.array(commonFluxes[key])

a = list()
for key in commonFluxes.keys():
    a.append(commonFluxes[key])
a = np.array(a)

meanFlux = np.mean(a,axis=0)
diffFluxes = dict()
b = list()
for key in commonFluxes.keys():
    diffFluxes[key] = (commonFluxes[key] - meanFlux)
    b.append((commonFluxes[key] - meanFlux))
b = np.array(b)

```

6 Show the spectrum of arc lines

Dotted line indicates the common wavelength. Some lines are not detected in some fibers and some lines are not listed in arc line list.

```

In [7]: butler = dafPersist.Butler('/gpfs01/pfs/LAM/rerun/calibs/arc/Xenon')
        visit = 17533
        dataId = {'visit': visit, 'arm': 'r', 'spectrograph': 1}
        arm = butler.get('pfsArm', dataId)
        fig, ax = plt.subplots(2, 1, figsize=(12,8))

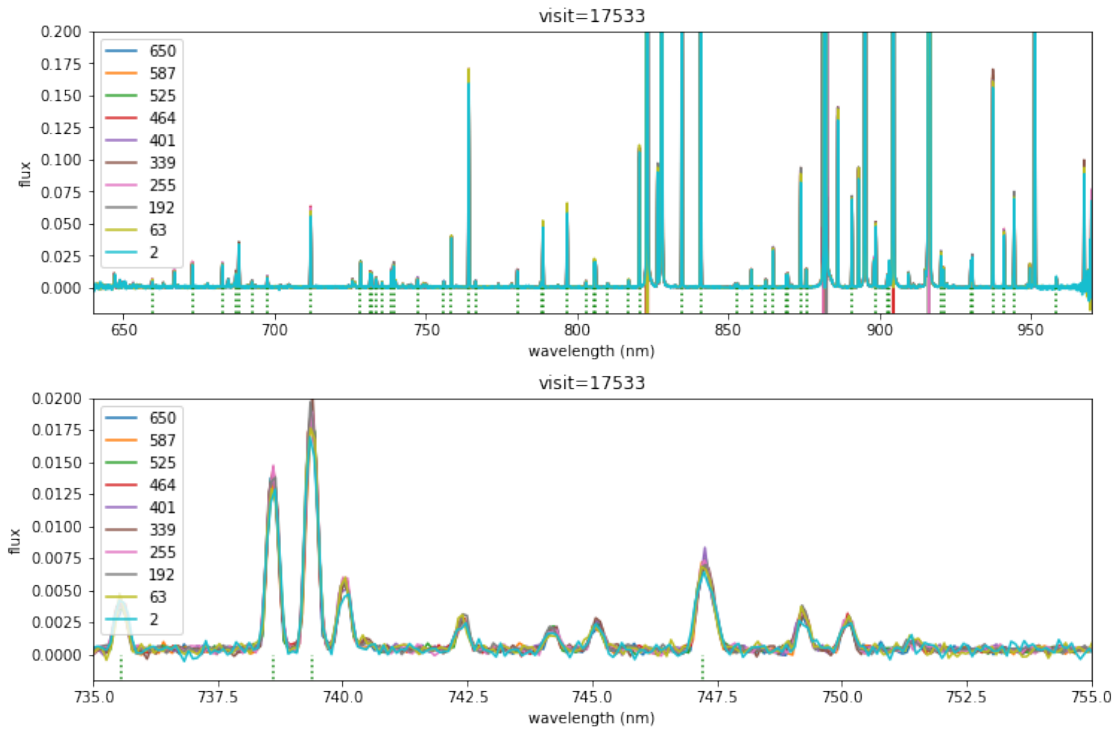
def plot(iax, xmin, xmax, ymin, ymax):
    for i in range(arm.numSpectra):
        w = arm.wavelength[i]
        f = arm.flux[i]
        ax[iax].plot(w, f, label='%d' % (arm.fiberId[i]))
    for wl in commonWL:
        ax[iax].plot([wl,wl], [-0.02, 0.0], 'g:')
    ax[iax].set_xlim(xmin, xmax)
    ax[iax].set_ylim(ymin, ymax)
    ax[iax].set_title('visit=%d' % (visit))
    ax[iax].set_xlabel('wavelength (nm)')
    ax[iax].set_ylabel('flux')
    ax[iax].legend(loc='upper left')

```

```

plot(0, 640, 970, -0.02, 0.2)
plot(1, 735, 755, -0.002, 0.02)
plt.subplots_adjust(hspace=0.3)
plt.show()

```

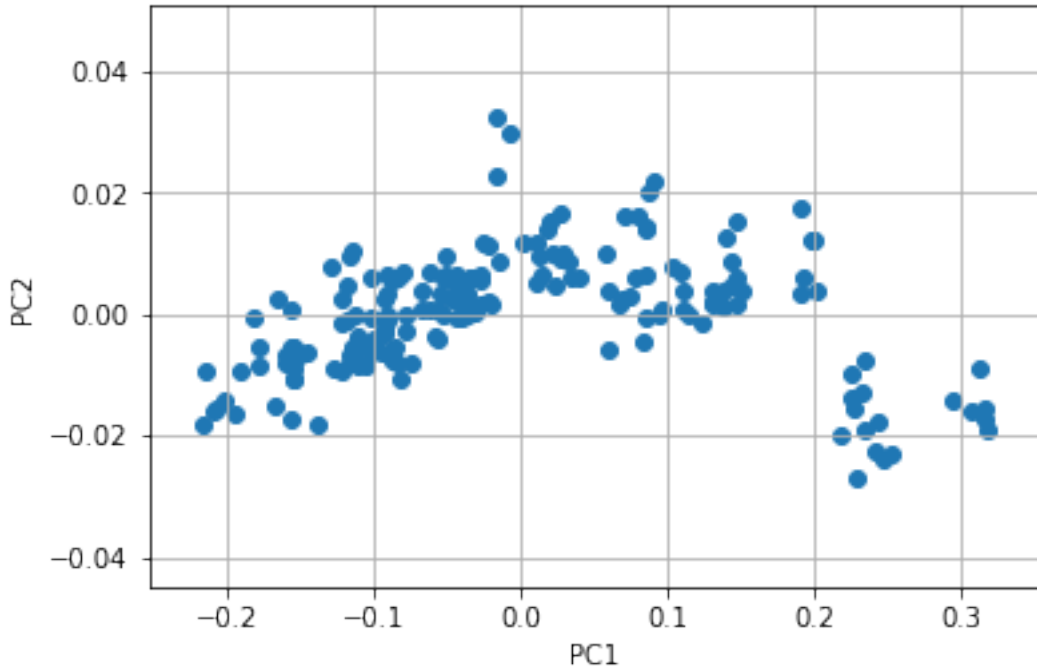


7 PCA analysis for measured fluxes of arc lines

```

In [8]:  pca = PCA(n_components=5)
         pca.fit(a)
         feature = pca.transform(a)
         plt.scatter(feature[:,0], feature[:,1])
         plt.grid()
         plt.xlabel("PC1")
         plt.ylabel("PC2")
         plt.show()

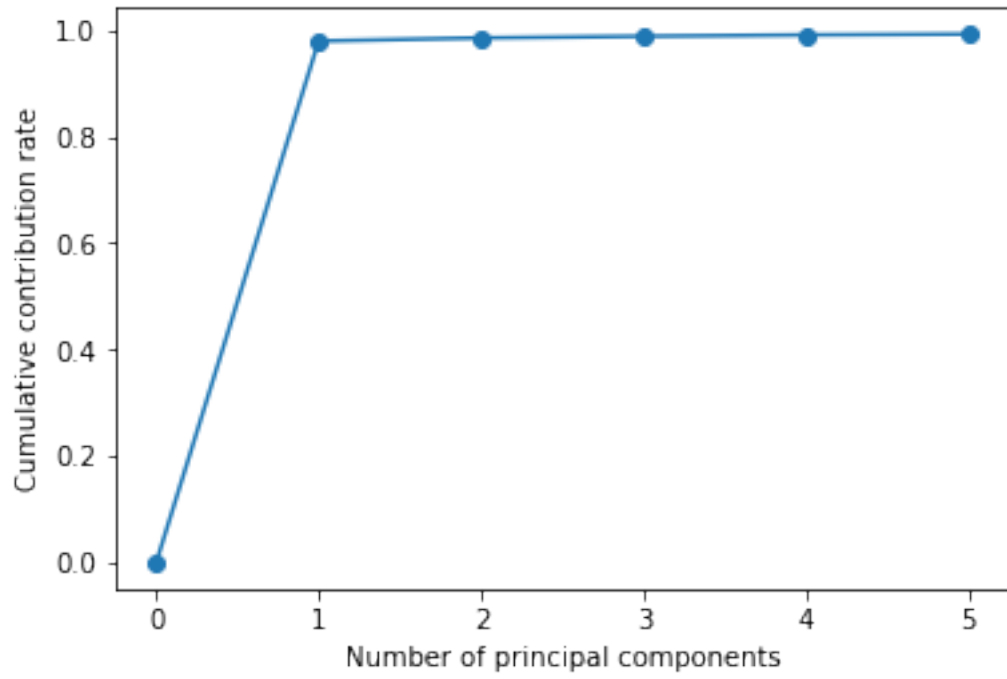
```



8 Cumulative contributions

The first component (probably overall brightness) carries almost all contribution. Is arc data appropriate to characterize using fluxes as a preparation of sky line characterization?

```
In [9]: plt.plot([0] + list(np.cumsum(pca.explained_variance_ratio_)), "-o")
plt.xlabel("Number of principal components")
plt.ylabel("Cumulative contribution rate")
plt.show()
```



In []: