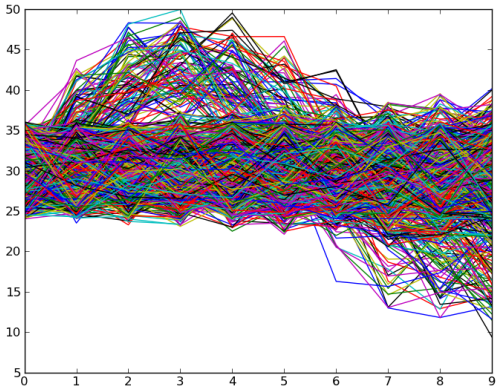
SVD ile Zaman Serisi Kumeleme

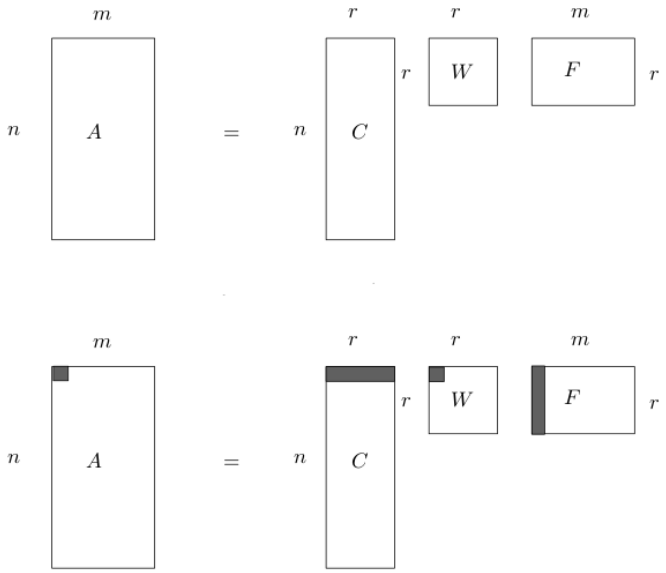
Tekil Deger Ayristirma (Singular Value Decomposition -SVD-) ile bir veri madenciligi ornegi gorecegiz. Ornek olarak [1] adresinde tarif edilen / paylasilan zaman serisini kullandik. Once veriyi grafikledik,



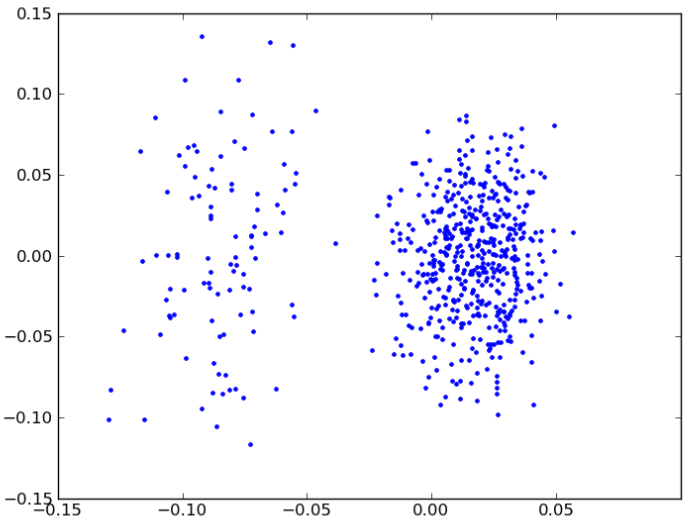
Verinin tamami kullanilmadi, serinin ilk 10 noktasini aldik, ve grafige bakinca iki tane ana seri oldugunu goruyoruz.

```
import numpy as np
from pylab import *

data = np.genfromtxt("synthetic_control.data", dtype=float)

print data.shape

for t in data[:,0:10]:
    plot(t); hold(True)

show()
```

Peki bu serileri nasil otomatik olarak kumeleyerek bulurduk / birbirinden ayirtederdik? *Lineer Cebir Ders 29*'da SVD'nin matematigini isledik. SVD bir matris A uzerinde ayristirma yapar, ve A herhangi boyutta, turde bir matris olabilir.

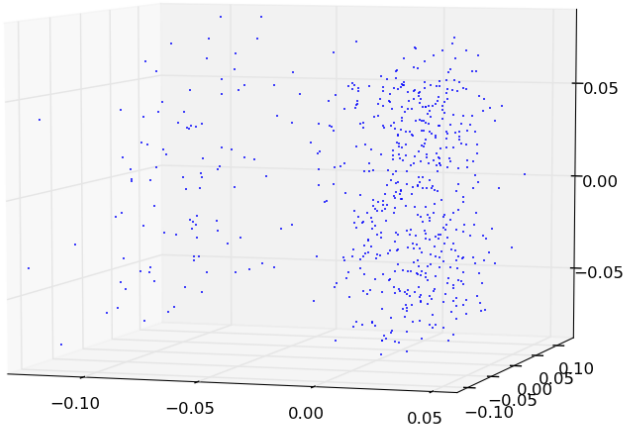


Ayristirmanin $A = CWF$ sonucunu verir, burada C, ana matris ile ayni miktarda satira sahiptir, F ayni miktarda kolona sahiptir. Ayristirma sonrasi A'nin kertesi (rank) ortaya cikar, eger tum A kolonlari birbirinden bagimsiz ise, o zaman $r = m$ olacaktir, ama kolonlarin bazilari mesela ayni olcumu degisik katlarda tekrarliyor ise, o zaman matriste tekillik vardir, ve bu durumda $r < m$ olur, ve ortadaki W matrisi $r \times r$ oldugu icin beklenenden daha ufak boyutlarda olabilir.




```
import scipy.linalg as lin
import numpy as np
from pylab import *

data = np.genfromtxt("synthetic_control.data", dtype=float)

# before norm, and take only 10 data points
data = data[:,0:10]
```

```
print data.shape

# show the mean, and std of the first time series
print data[0,:]
print np.mean(data[0,:], axis=0)
print np.std(data[0,:], axis=0)

# normalize
data -= np.mean(data, axis=0)
data /= np.std(data, axis=0)

# after norm
print data[0,:]

u,s,v = lin.svd(data, full_matrices=False)
print 'svd'
print u.shape
print s
print v.shape

plot(u[:,0], u[:,1], '.')

print u[:,0] < -0.025

show()
```




```
from mpl_toolkits.mplot3d import Axes3D
import scipy.linalg as lin
import numpy as np
from pylab import *

data = np.genfromtxt("synthetic_control.data", dtype=float)

data = data[:,0:10]
```

```
print data.shape

data -= np.mean(data, axis=0)
data /= np.std(data, axis=0)

u,s,v = lin.svd(data)
print 'svd'
print u.shape
print s
print v.shape

fig = plt.figure()
ax = Axes3D(fig)
ax.plot(u[:,0], u[:,1], u[:,2],',', zs=0, zdir='z', label='zs=0, zdir=z')

show()
```

[1] http://kdd.ics.uci.edu/databases/synthetic_control/synthetic_control.data.html