In [35]:

```
# importing modules
import numpy as np
import numpy.linalg as LA
```

In [16]:

```
f = open("/home/tandon/IIIT-H/3rd/SMAI/dorothea/dorothea_train.data")
X = np.zeros((800, 100000))
row = 0
for line in f:
    for token in line.split():
        idx = int(token)
        X[row, idx-1] = 1
#         print(idx)
    row += 1
print('Done')
```

```
Done
```

In [17]:

```
print(X[1,306])
```

```
1.0
```

In [26]:

```
# calculate mean vector
m = X.mean(0)
print(m)
m.shape
```

```
[ 0.0125   0.       0.00125 ...,  0.0125   0.00625  0.035  ]
```

Out[26]:

```
(100000,)
```

In [31]:

```
At = X - m
```

In [32]:

```
A = At.T
A.shape
```

Out[32]:

```
(100000, 800)
```

In [33]:

```
K = At.dot(A)
```

In [43]:

```
K.shape
eigen_values, eigen_vectors = LA.eig(K)
```

In [46]:

```
print('Eigen Values shape', eigen_values.shape)
print('Eigen vectors shape', eigen_vectors.shape)
```

```
Eigen Values shape (800,)
Eigen vectors shape (800, 800)
```

In [49]:

```
# for K = 100, need last 100 eigen vectors
e100 = eigne_vectors[:, 700:]
e100.shape

e500 = eigne_vectors[:, 300:]
e500.shape
```

Out[49]:

```
(800, 500)
```

In [50]:

```
# calculating final eigen vector for k = 100
axis100 = A.dot(e100)
axis100.shape

axis500 = A.dot(e500)
axis500.shape
```

Out[50]:

```
(100000, 500)
```

In [ ]:

```

```