Asmt 7: Dimensionality Reduction

Han Ambrose
Turn in through Canvas by 2:45pm:
Monday, April 8
100 points

1 Singular Value Decomposition (70 points)

First we will compute the SVD of the matrix A we have loaded

```
import numpy as np
from scipy import linalg as LA
U, s, Vt = LA.svd(A, full_matrices=False)
```

Then take the top k components of A for values of $k = 1$ through $k = 10$ using

```
Uk = U[:,:k)
Sk = S[:k,:k]
Vtk = Vt[:k,:]
Ak = Uk @ Sk @ Vtk
```

A (40 points): Compute and report the L_2 norm of the difference between A and Ak for each value of k using

```
LA.norm(A-Ak,2)
```

k	L2 Norm
1	106.8
2	98.93
3	93.82
4	75.57
5	62.99
6	61.57
7	27.68
8	26.45
9	26.27
10	24.6

Table 1: L_2 Norm of $A - Ak$

```
A = np.loadtxt('A.csv', delimiter= ',')
print(A.shape)
U, s, Vt = LA.svd(A, full_matrices=False)
print(U.shape, s.shape, Vt.shape)
#convert s to diagonal matrix
S = np.diag(s)

#Question 1A
for k in range(1,20):
  Uk = U[:,:k]
  Sk = S[:k,:k]
  Vtk = Vt[:k,:]
  Ak = Uk @ Sk @ Vtk
  print('k = ',k,'L2 norm difference is %.2f' % LA.norm(A-Ak,2))

#Question 1B
  if LA.norm(A-Ak,2) < 0.1*LA.norm(A,2):
    print('k= %d',k);
```

B (10 points): Find the smallest value k so that the L_2 norm of `A-Ak` is less than 10% that of `A`; k might or might not be larger than 10.

L2 norm of A is 123.85 so 10% of that is 12.4.
Just looking at the table above, we have to continue testing for $k > 10$. We found that the smallest value k so that the L_2 norm of `A-Ak` is less than 10% is 19

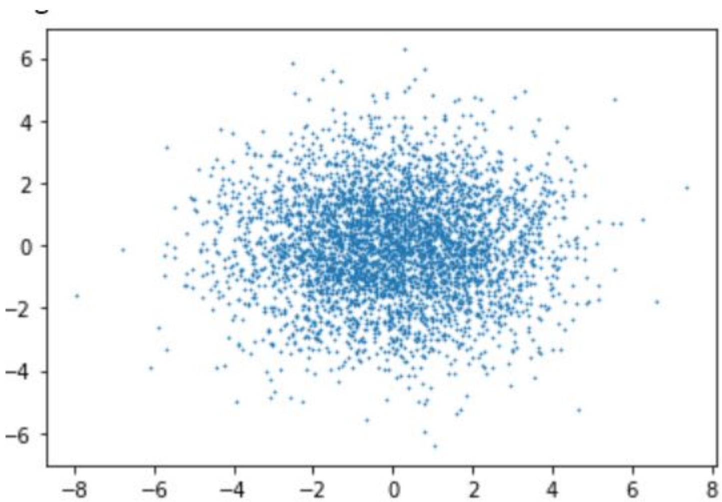
C (20 points): Treat the matrix as 5000 points in 40 dimensions. Plot the points in 2 dimensions in the way that minimizes the sum of residuals squared, and describe briefly how you did it.

We need to find the subspace F to minimize:

$$\|A - \pi_F(A)\|_F^2 = \sum_{i=1}^{n} \|a_i - \pi_F(a_i)\|^2$$

First we need to make sure to restrict the subspace V_k to go through the origin. By using centering matrix $C_n = I_n - \frac{1}{n}11^T$. Then we have the new centered maxtrix $A' = C_n A$
Then we run $SVD(A')$. Next, the first 2 singular vectors $\{v_1, v_2\}$ were used to reduce the dimension from 20 dimension to 2 dimension. Since the first two singular vectors represent eigenvectors, this projection will result in the least sum of residuals squared.

The figure below shows that most of the dots are cetered at through the origin.




```
n = 4000
oneoneT = np.ones((n, n))
iden = np.identity(n)
Cn = iden - (1/n)*(oneoneT)
print(Cn.shape)
A_center = Cn @ A

U_new, s_new, Vt_new = LA.svd(A_center, full_matrices=False)

#reduce to 2 dimension : Vt 20 x 2
V_2 = Vt_new[:2,:].transpose() # the first 2 right singular values

# Projection of all points on the eigen vectors
PointsToPlot = A_center.dot(V_2)
plt.scatter(PointsToPlot[:,0], PointsToPlot[:,1], s=0.5)
plt.figure(figsize=(20,10))
```

2 Frequent Directions and Random Projections (30 points)

Use the stub file `FD.py` to create a function for the Frequent Directions algorithm (**Algorithm 16.2.1**). Consider running this code on matrix `A`.

A (30 points): Measure the error $\max_{\|x\|=1} |\|Ax\|^2 - \|Bx\|^2|$ as `LA.norm(A.T @ A - B.T @ B)`

- How large does `l` need to be for the above error to be at most $\|A\|_F^2/10$?

  Using the algorithm below, we were able to find the error for each l. $\|A\|_F^2/10 = 6463$, so $l \geq 7$ for the error to be at most $\|A\|_F^2/10$

  Set B all zeros $(2\ell \times d)$ matrix.
  for rows (i.e. points) $a_i \in A$ **do**
  Insert a_i into a zero-valued row of B
  if (B has no zero-valued rows) **then**
  $[U, S, V] = \texttt{svd}(B)$
  Set $\delta_i = \sigma_\ell^2$ # the ℓth entry of S
  Set $S' = \text{diag}\left(\sqrt{\sigma_1^2 - \delta}, \sqrt{\sigma_2^2 - \delta}, \ldots, \sqrt{\sigma_{\ell-1}^2 - \delta}, 0, \ldots, 0\right)$.
  Set $B = S'V^T$ # the last rows of B will again be all zeros
  return B

ℓ	error
1	15,339
2	15,318
3	15,313
4	15,264
5	12,219
6	9,279
7	6,375
8	4,504
9	2,894
10	1,934

- How does this compare to the theoretical bound (e.g. for $k = 0$).

  The theoretical bound is $0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \frac{\|A-A_k\|_F^2}{l-k}$. In this case, $k = 0$, the bound is $\frac{\|A\|_F^2}{10}$. So $l = 10$. Since $l = \frac{1}{\epsilon}$, $\epsilon = 0.1$
- How large does `l` need to be for the above error to be at most $\|A - A_k\|_F^2/10$ (for $k = 2$)?

  $0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \frac{\|A-A_k\|_F^2}{l-k}$. So $\frac{1}{l-k} = \frac{1}{10}$, $k = 2$, so $l = 10 - 2 = 8$