Asmt 6: Dimensionality Reduction

Turn in through GradeScope by 1:00pm:
Wednesday, April 2
100 points

Overview

In this assignment you will explore regression techniques on high-dimensional data.

You will use two data sets for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/DM/A6/A.csv`
- `http://www.cs.utah.edu/~jeffp/teaching/DM/A6/D.csv`

For python, you can use the following approach to load the data:

```
df = pd.read_csv('A.csv')
```

As usual, it is recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory: *`http://www.cs.utah.edu/~jeffp/teaching/latex/`*

Click here for an example template specifically created for this assignment.

1 Singular Value Decomposition and PCA (60 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 (20 points): Compute and report the L_2 norm of the difference between `A` and `Ak` for each value of k (10 values) using

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

B (10 points): Find the smallest value k so that the L_2 norm of `A-Ak` is less than 5% that of the L_2 norm of `A`; k might or might not be larger than 10. Report (i) the L_2 norm for `A`, (ii) the L_2 norm for your choice of `A-Ak`, and (iii) your choice of k.

C (10 points): Plot the points in 2 dimensions by projecting `A` onto the top 2 right singular values.

D (10 points): Now repeat (**B**) for PCA. First center the data to get $\tilde{\texttt{A}}$, and then find the value k where the L_2 norm of $\tilde{\texttt{A}}$`-Ak` is less than 5% that of the L_2 norm of $\tilde{\texttt{A}}$.

E (10 points): Plot the points in 2 dimensions by projecting `A` onto the top 2 principal components.

2 Multidimensional Scaling (40 points)

You will apply multidimensional scaling on an all-pairs distance among US airports, stored as matrix `D`.

A (10 points): Transform the matrix into a $D^{(2)}$ matrix where each element is squared. Report the Frobenius norm of $D^{(2)}$.

B (10 points): Double center the matrix so $M = -\frac{1}{2}C_n D^{(2)} C_n$, and report the Frobenius norm of M.

C (20 points): Plot the data in 2 dimensions on the top 2 eigenvectors of M.

3 BONUS (3 points):

Create another `l x d` matrix B, but using random projections. You can do this by creating an `l x n` matrix `S`, and letting `B = SA`. Fill each entry of `S` by an independent normal random variable $S_{i,j} = \frac{1}{\sqrt{1}} N(0,1)$.

Empirically estimate how large should `l` be in order to achieve $\max_{\|x\|=1} |\|Ax\|^2 - \|Bx\|^2| \leq \|A\|_F^2/20$. To estimate the relationship between `l` and the error in this randomized algorithm, you will need to run multiple trials. Be sure to describe how you used these multiple trials, and discuss how many you ran and why you thought this was enough trials to run to get a good estimate.

4 BONUS (10 points):

Professor Phillips recently found the following method for distance metric learning in a paper, without much explanation. Let $C = \{x_i, x_i'\}$ be a set of n_C points we would like to be close from each other, each x_i in $\mathbb{R}^d$. Let $F = \{x_j, x_j'\}$ be a set of n_F points we would like to far from each other, each x_j in $\mathbb{R}^d$. Define an $d \times d$ matrix

$$M = \left(\alpha I + \frac{\beta}{n_C} \sum_{\{x_i, x_i'\} \in C} (x_i - x_i')(x_i - x_i')^T - \frac{\gamma}{n_F} \sum_{\{x_j, x_j'\} \in F} (x_j - x_j')(x_j - x_j')^T \right)^{-1},$$

for some values of $\alpha, \beta, \gamma > 0$.

Describe why, or under which scenarios, or for which α, β, γ that the resulting Mahalanobis distance $\mathsf{d}_M(p,q) = \sqrt{(p-q)^T M (p-q)}$ respects the close and fair input pairs. Or show that it does not generally work (very well). Maybe trying it out on some data would be a good place to start.