CSE 3380 – Homework #10

Assigned: Thursday, April 19, 2018
Due: Thursday, April 26, 2018 at the end of class

Note the following about the homework:

1. You must show your work to receive credit.
2. If your submission has more than one page, staple the pages. If I have to staple it, the cost is 10 points.

Assignment:
Process

1. (hand solution) Find the Singular Value Decomposition (SVD) of A. Use the reduced version if the situation allows it.
$$A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 2 \end{bmatrix}$$

2. (hand solution) Find the Singular Value Decomposition (SVD) of A. Use the reduced version if the situation allows it.
$$A = \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix}$$

Theory

3. We have seen that when given an $m \times n$ matrix A and an $n \times p$ matrix B, we can produce AB using dot products (i.e., inner products). If we think of A has consisting of n column vectors and B as consisting of n row vectors, we can produce AB using the outer product.

   That is, if
$$A = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \\ | & | & & | \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} - & \vec{b}_1 & - \\ - & \vec{b}_2 & - \\ & \vdots & \\ - & \vec{b}_n & - \end{bmatrix}$$
   then
$$AB = \sum_{i=1}^{n} \vec{a}_i \vec{b}_i$$
   Pay attention to what this says. This is the product of the first column of A with the first row of B plus the product of the second column of A with the second row of B and so forth.

   Let
$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} e & f & g \\ h & i & j \end{bmatrix}$$
   Produce AB using

(a) normal matrix multiplication

(b) the outer product.

In both cases, show enough work to make it possible to see how the two approaches work and why they must produce the same result.

Applications

4. (Python solution) In order to use the normal equations to find a least squares solution to $A\vec{x} \cong \vec{b}$, we must be able to produce $(A^TA)^{-1}$. This is only possible if A is of full column rank.

   One of the applications of the SVD is to produce a least squares solution to a system of equations regardless of A's shape or rank. Another is to determine the effective rank of a matrix. We will use both of these for this problem.

   (a) On the course website is a file, `svdMain.py`, that contains an $m \times n$ matrix A and a vector b with m elements. These are given to a function with the signature

   ```
   x1, norm1, x2, norm2, erank, x3, norm3 = svdStudent(A, b)
   ```

   The return value are described below. Write the function `svdStudent()`, which will be placed in the file `svdStudent.py`. Your function should not be hard-coded to this particular data; use what is passed in.

   i. Solution 1: find `x1` using least squares. Also produce `norm1`, which is the norm of the residual using this value of $\vec{x}$.

   ii. Solution 2: find the SVD of A using numPy's `svd()` function. Then find `x2` using

   $$\vec{x2} = \sum_{i=1}^{n} \frac{\vec{u}_i^T\vec{b}}{\sigma_i}\vec{v}_i$$

   where n is the number of columns of A. Also produce `norm2`, which is the norm of the residual using this value of $\vec{x}$.

   iii. Use the SVD of A to find the effective rank of A, `erank`. We will consider the effective rank to be the number of singular values with values greater than or equal to 0.001. Note that if we have a matrix S, then numPy allows comparisons like

   ```
   S >= 0.001
   ```

   which can then be used to automatically count the number of matrix values greater than or equal to a specific value.

   iv. Solution 3: find `x3` using

   $$\vec{x3} = \sum_{i=1}^{r} \frac{\vec{u}_i^T\vec{b}}{\sigma_i}\vec{v}_i$$

   where r is the effective rank of A that you found in part 4(a)iii. This sum assumes that the first r singular values are those that are nonzero (or at least greater than our threshold). Also produce `norm3`, which is the norm of the residual using this value of $\vec{x}$.

v. The true value of $\vec{x}$ is $[1, -2, 3, -4, 5, -6, 7]^T$. For yourself, think about which of the three solutions for $\vec{x}$ that you found is the closest? Does this tell you anything about whether you should blindly rely on the norm of the residual as the way to determine the "best" solution?

5. **(CS application: artificial intelligence)** (Python solution) A popular task in artificial intelligence is to classify things; popular choices are images and documents. Many methods for performing classification have been developed; for this we will use the SVD of a matrix.

The process is as follows:

(a) Training samples from each of `c` classes of objects are chosen, with each object represented as a vector in $\mathbb{R}^n$ where n is the number of attributes describing each object (for example, pixels for images).

(b) For each class, create an $n \times q$ matrix where q is the number of training samples for that class.

(c) Find the SVD of each of these training matrices.

(d) Given a test sample z (which needs to be represented as a column vector), for each class calculate

$$||(I - U_k U_k^T)z||_2 \qquad \text{(Eq 1)}$$

where U_k is a matrix formed from the first k columns of the U matrix of a specific class (these should correspond to the largest singular values).

(e) Classify z as belonging to the class whose norm produced the smallest value.

Problem:

On the course website is a file, `aiSVDmain.py`, that contains an $m \times n$ matrix A and a $m \times 1$ vector b. These are given to a function with the signature

```
Ukp, Ukx, np1, nx1, np2, nx2 = aiSVDstudent(k, pClass, xClass, test1, test2)
```

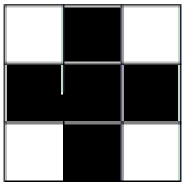
where

- `Ukp` is the first k columns of the U matrix from the SVD of the plus class training matrix
- `Uxp` is the first k columns of the U matrix from the SVD of the X class training matrix
- `np1` is the result of `Eq 1` for the plus class and test sample 1
- `nx1` is the result of `Eq 1` for the X class and test sample 1
- `np2` is the result of `Eq 1` for the plus class and test sample 2
- `nx2` is the result of `Eq 1` for the X class and test sample 2

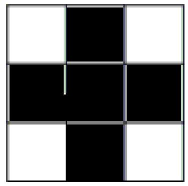
Write the function `aiSVDstudent()`, which will be placed in the file `aiSVDstudent.py`. Your function should not be hard-coded to this particular data; use what is passed in.

General requirements about the Python problems:

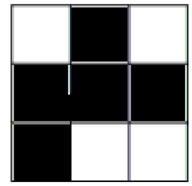
a) As a comment in your source code, include your name.



(a) plus_1

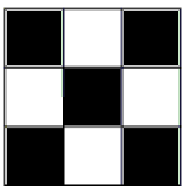


(b) plus_2

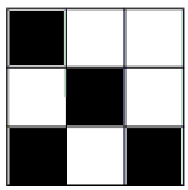


(c) plus_3

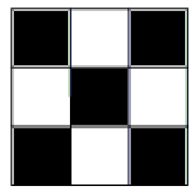
Figure 1: Training samples for class `plus`



(a) x_1

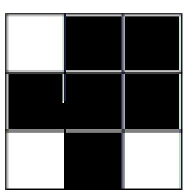


(b) x_2

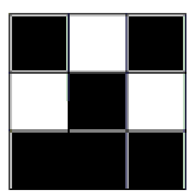


(c) x_3

Figure 2: Training samples for class `X`



(a) t_1



(b) t_2

Figure 3: Test samples

b) The Python program should do the work. Don't perform the calculations and then hard-code the values in the code or look at the data and hard-code to this data unless instructed to do so.

c) Your function should use the data passed to it and should work if I were to change the data.

d) The program should not prompt the user for values or read from files unless instructed to do so.

e) I don't want your function to cause anything to print; leave it to the `XXXmain.py` file to do the printing of what is returned.

To submit the Python portion, do the following:

a) Create a directory using your net ID in lowercase characters. This should be something of the form `abc1234`.

b) Place your .py files in this directory.

c) Zip the directory, not just the files within the directory. You must use the zip format and the name of the file, assuming your net ID is `abc1234`, will be `abc1234.zip`.

d) Upload the zip'd file to Blackboard.

Review Questions

These are not for credit, but instead are intended to test your understanding of the concepts. You should be able to answer these without simply regurgitating equations.

1. Given a system of equations, how is the SVD used to find $\vec{x}$?
2. How is using the outer product to produce the SVD of a matrix useful?

References

[Eld07] Lars Eldén. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007.