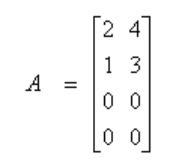
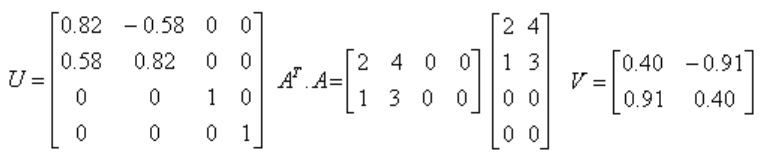
SVD:

1. **For any given matrix, is the SVD necessarily unique?**
   * **Answer: No.**
2. **Name three applications of the SVD.**
   * **Possible Answers: Computing the pseudoinverse, matrix approximation, and determining the rank, range and null space of a matrix.**
3. **Provide the definition of a singular value.** 
   * **Answer: The singular values are square roots of eigenvalues from AAT or ATA.**
4. **Ca**lculate the SVD of the matrix below.



**Answer:**



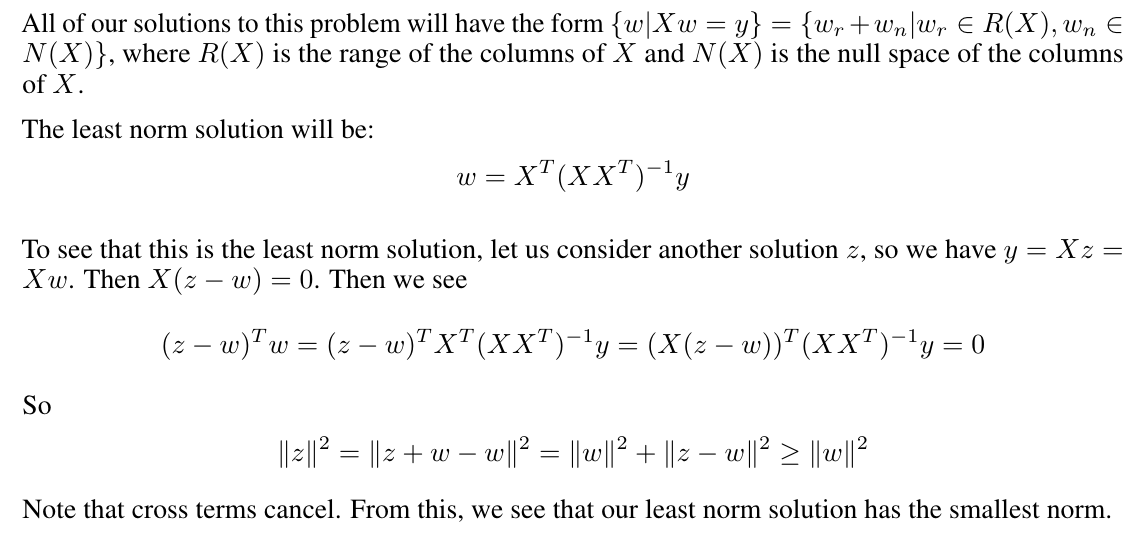
PCA

1. Is this statement true or false? **Even if all the input features are on very similar scales, we should still perform mean normalization (so that each feature has zero mean) before running PCA.**
   1. **Answer: True.**
2. **If our input features are on very different scales, what should we do before applying PCA?**
   1. **Answer: We should perform feature scaling.**

Min Norm

1. **Prove that the Moore-Penrose inverse for an underdetermined system of linear equations gives you the minimum norm solution.**

* Answer:



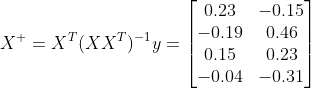
1. **Find the minimum norm solution for the following system:**



* Answer:

, , , where 

Solve for Moore-Penrose inverse:



Then 

Code:

|  |
| --- |
| X = np.array([[2, -1, 2, -1], [0, 1, 1, -1]])  y = np.array([[6, 1]]).reshape((2,1))  print('X:\n', X); print('\ny:\n', y)  X\_plus = np.linalg.pinv(X)  print('\nX\_plus:\n', X\_plus)  w = X\_plus.dot(y)  print('\nMinimum norm w:: \n', w.round()) |
| X:  [[ 2 -1 2 -1]  [ 0 1 1 -1]]  y:  [[6]  [1]]  X\_plus:  [[ 0.23076923 -0.15384615]  [-0.19230769 0.46153846]  [ 0.15384615 0.23076923]  [-0.03846154 -0.30769231]]  Minimum norm w::  [[ 1.]  [-1.]  [ 1.]  [-1.]] |

BCI

1. **What are some of the applications of brain computer interfaces?**

* Answer:
* Understanding nervous system
  + Neuronal signals/spiking when given a stimuli (sensory, visual, etc.)
  + Neural signals/spiking with a given effector signal (movement intention, speech intention, etc)
* Augmenting Human Functions
  + Controlling computers
  + Controlling robots
  + Etc.
* Aiding in neurological diseases
  + Deep brain stimulation
  + Closed loop stimulators
  + Sensory purposes (hearing, sight)
* The sky's the limit for what one can dream about the applications of BCI!

1. **You have a matrix X, representing n neuron spikes in a window of d samples. With this matrix, describe how you will split X into training and test sets. Then, describe how you will use PCA to identify neuronal clusters and classify neurons in your test set.**

* Answer:
  + With your matrix X, split into test and training sets (taking about 30% of the data for example for testing). This means X\_train will be of size .7n x d, and X\_test will be of size .3n x d.
  + The d represents the signals of the waveforms. We want to use PCA to project it into a lower dimensional space, say 2 dimensions, in order to perform waveform clustering.
  + PCA:
    - Take the mean waveform for X\_train, and subtract: X\_train’ = X\_train - mean(X\_train)
    - Perform SVD => X\_train’ = U S V^T, where S is sorted from highest singular value to lowest. The first 2 column vectors of V represent our PCA basis.
  + Project X\_train onto the first 2 columns of V, and perform clustering to identify neuronal clusters. (K-means is one clustering algorithm where we can iteratively assign centroids and classes until the distance from each point to the assigned centroid is minimized)
  + For the data in X\_test, subtract off mean(X\_train), then project X\_train onto the first 2 vectors of V (take the dot product). With this projection, you can take the nearest centroid as the class of our new neuron waveform. With known classes y\_train and y\_test, we can compute the accuracy of our predictions.
  + In actuality, we may not have y\_train and y\_test, and as such rely upon our unsupervised clustering algorithm to give us classes of neurons.