SVD

PCA

Min Norm

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

     To see that this is the least norm solution, let us consider another solution z, so we have $y = Xz = Xw$. Then $X(z - w) = 0$. Then we see

     $$(z-w)^T w = (z-w)^T X^T (XX^T)^{-1} y = (X(z-w))^T (XX^T)^{-1} y = 0$$

     So

     $$\|z\|^2 = \|z + w - w\|^2 = \|w\|^2 + \|z - w\|^2 \geq \|w\|^2$$

     Note that cross terms cancel. From this, we see that our least norm solution has the smallest norm.

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

   $$\min_{w \in \mathbb{R}^4} ||w|| \quad \text{s.t.} \quad \begin{aligned} 2w_1 - w_2 + 2w_3 - w_4 &= 6 \\ w_2 + w_3 - w_4 &= 12 \end{aligned}$$

   - Answer:

   $X = \begin{bmatrix} 2 & -1 & 2 & -1 \\ 0 & 1 & 1 & -1 \end{bmatrix}$, $w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$, $y = \begin{bmatrix} 6 \\ 1 \end{bmatrix}$, where $Xw = y$

   Solve for Moore-Penrose inverse:

   $$X^+ = X^T (XX^T)^{-1} y = \begin{bmatrix} 0.23 & -0.15 \\ -0.19 & 0.46 \\ 0.15 & 0.23 \\ -0.04 & -0.31 \end{bmatrix}$$

   Then $w = X^+ y = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}^T$

   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!
2. **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.