Problem 1: Principal Component Analysis

(20 points)
In this question, you will build a spam filter using the K-Nearest Neighbour algorithm like the one you built on Homework 8. This time, you will be using Principal Component Analysis (PCA) to transform the e-mail data from 2 dimensions (and a label) to 1 dimension (and a label). Recall that each e-mail has two features: number of words in the email and total occurrences of the spam words “credit” and “dollars”. The following graph shows the training data where filled circles are spam e-mails and unfilled circles are non-spam emails. The x-axis (Total words count) has been scaled for your convenience.

We have found the first principal component: \((\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})\). Therefore the linear subspace that PCA will project the data to is the line \(y = x\). (Assume that we don’t need to do any centering of the data.)

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Graph_of_E-Mail_Data.png}
\caption{Graph of E-Mail Data}
\end{figure}

a. Draw the first principal component as a line \((y = x\) in this case) on the graph. Also draw the data points projected to this subspace with lines connecting them to the corresponding original data points.

Solution:
b. How would 1-Nearest Neighbour algorithm classify an e-mail with 21 spam words in a total of 100 ($x=5$) words? Explain your answer.

Solution: Spam. The nearest point is (5, 25) and labeled Spam.

c. How would 1-Nearest Neighbour algorithm classify the same e-mail given in the previous question after transforming the data to 1-dimension? Explain your answer.

Solution: Non-spam. After projection, the nearest neighbour to the test data is a non-spam e-mail.

d. Are your answers same for part b and c? Why or why not?

Solution: No, they are different because we lost some information after projection.

Problem 2: Exact Recovery of a Linear Compression Scheme

In this exercise we show that in the general case, exact recovery of a linear compression scheme is impossible.

a. Let $A \in \mathbb{R}^{n,d}$ be an arbitrary compression matrix where $n \leq d - 1$. Show that there exists $u, v \in \mathbb{R}^d$, $u \neq v$, such that $Au = Av$.

Hint: Show that there exists $u \neq 0, v = 0$ such that $Au = Av = 0$.

Hint: Consider using the rank-nullity theorem.

Solution: By the rank-nullity theorem, $\text{rank}(A) + \text{nullity}(A) = d$. Because $n < d$, $\text{rank}(A) \leq n$. It follows that $\text{nullity}(A) = d - \text{rank}(A) \geq d - n \geq d - (d - 1) = 1$. Therefore, there must exist $u \neq 0$ s.t. $Au = 0$ because the nullity of $A$ is positive. It follows that $Au = Av = 0$ where $v = 0$.

b. Conclude that exact recovery of a linear compression scheme is impossible.

Solution: If $Au = Av$, then we cannot distinguish mappings of $u$ from mappings of $v$. Therefore, we cannot exactly recover the original data, meaning that exact recovery of a linear compression scheme is impossible.
Problem 3: Limitations of PCA (20 points)

Consider the following two-dimensional dataset:

a. Describe one type of machine learning model that would classify this data well. Explain your reasoning.

**Solution:** A halfspace classifier would classify this data well because the data is linearly separable across the line $y = x$.

b. Draw the first principal component on the graph above. If PCA was used to reduce this data to one dimension, would the machine learning model from part (a) still classify the data well? Why or why not?

**Solution:** The first principal component is along the line $y = x$. If the data was projected into this subspace, there would no longer be any clear classification trend in the data, meaning that the halfspace classifier from part (a) would no longer classify the data well.

c. Is it possible to project the above data into a one-dimensional linear subspace in which the data remains linearly separable? If so, draw the subspace on the graph above. If not, explain your reasoning.

**Solution:** If the data is projected into the subspace $y = -x$, it will remain linearly separable.

d. What does this tell example tell you about the limitations of PCA when used to pre-process data before classification?

**Solution:** PCA is not always helpful. For some datasets like the one above, PCA can reduce information and significantly hurt the performance of a classifier.