Homework 4: PCA, SVM
CPE 490/590 ST

Instructor: Rahul Bhadani

Due: April 2, 2025, 11:59 PM
80 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

Submission instruction:

Submission instruction for this homework supersedes one mentioned in the Syllabus.

This homework requires all answers recorded in a single `.ipynb` Python notebook. You may upload hand-written PDF for the theory portion. You can use a combination of text cell (i.e. markdown formatted cell) and code cell to provide your answer. To add equations you should be able to use Latex syntax in the text cells of your Python notebook. As a part of your submission, you must provide executed notebook with code, text, and outputs. Alternatively, you can also provide a url (whose permission you must change to 'anyone with link can view') of your Python notebook from Google Colab. The naming convention for your notebook should follow the format {firstname_lastname}_CPE 490/590 ST_hw03.ipynb. For example, if your name is Sam Wells, and you are enrolled in CPE 490 your file name should be sam_wells_CPE490_hw03.ipynb.

Please refer to https://github.com/rahulbhadani/CPE490_590_Sp2025/blob/master/Code/ for hands-on. Theory

1 Relation between Constraint Optimization problem and SVM (10 points)

Write down the formulation (i.e. mathematical equation) of the constraint optimization problem used in Support Vector Machine (SVM).

2 Distance between Hyperplanes (5 points)

Consider two hyperplanes:

$$H_1 : \mathbf{w}^\top \mathbf{x} = 52$$

and

$$H_2 : \mathbf{w}^\top \mathbf{x} = 64$$

where w is unknown.

What is the distance between the two hyperplanes H_1 and H_2?

3 Overfitting in SVM (5 points)

What strategy (such as a change in SVM formulation) would you take to avoid overfitting in the case of a Support Vector Machine?

Practice

4 Principal Component Analysis on Diabetes Dataset (20 points)

Download the diabetes dataset from the following link: https://github.com/rahulbhadani/CPE490_590_Sp2025/tree/master/Data/Diabetes. Your task is to perform dimensionality reduction using Principal Component Analysis (PCA) and answer the following questions **(2.5 points each)**.

1. **Principal Component Analysis (PCA) with Two Components:** Perform PCA on the dataset, selecting only two principal components. Report the variance explained by these two components.
2. **2D Plot Analysis:** Generate a 2D plot using the two principal components obtained from the previous step. Analyze the plot and discuss whether the two classes (0 and 1 in the 'Outcome' column of the Diabetes dataset) are distinguishable.
3. **PCA with Three Components:** Now, perform PCA again, but this time select three principal components. Report the variance explained by each component and discuss the contribution of the additional third component.
4. **3D Plot Analysis:** Generate a 3D plot using the three principal components obtained from the previous step. Analyze the plot and discuss whether the two classes are more distinguishable in the 3D plot compared to the 2D plot.

5 Flavors of Support Vector Machine (20 points)

1. Create a synthetic dataset consisting of two separable class using the following code:

```
from sklearn.datasets import make_classification
import pandas as pd

# Set random seed for reproducibility
seed = 200

# Generate synthetic dataset
X, y = make_classification(n_samples=500, # Number of samples
                            n_features=2, # Number of features
                            n_redundant=0, # Number of redundant features
                            n_informative=2, # Number of informative features
                            n_clusters_per_class=1, # Number of clusters per class
                            n_classes=2, # Number of classes (binary)
                            flip_y=0.00, # Noise level
```

```
                                class_sep=2.5, # Factor separating the classes
                                random_state=seed)
```

Make sure your seed value stays at 200. Create a linear support vector machine classifier with $C = 1e6$ (that is clear separation), and visualize the decision boundary **(5 points)**.

2. Change the noise level so that we can have some misclassification by setting `flip_y = 0.02` in `make_classification` function. Plot the synthetic data and write one statement on what you observe in the plot **(2 points)**.

3. Implement a Linear SVM classifier with $C = 1e6$ (that is clear separation) on data generated in the previous step. Make a comment on what you observe **(3 points)**.

4. Now add some fuzziness to the Linear SVM classifier by setting $C = 1$ (so that there is no overfitting). Comment what you observe **(5 points)**.

5. Set `class_sep = 15.` in `make_classification` function and regenerate the Synthetic dataset. Implement a nonlinear SVM using a polynomial kernel of degree 3. Choose coef0=1, and C=5. Comment what you observe. **(5 points)**.