Homework 5

In this homework, we will implement a Naive Bayes classifier and a KNN classifier.

You will be given two training datasets: `hw5_grass.txt` and `hw5_cat.txt`. Each dataset is stored as a data matrix of size 64×500. Let $\boldsymbol{x}_k^{(\text{cat})}$ be the k-th column of the `cat` matrix. The size of $\boldsymbol{x}_k^{(\text{cat})}$ is 64×1, and it represents a *feature vector* of a training sample. There are 500 of these training samples in each dataset.

In your repository, you will also find an image `cat_grass.png`. Load this image into Python using `matplotlib.pyplot`'s command `imread`. For your convenience, we have written a data pre-processing function `preprocess_data` in `helper.py` which you can input the image and extract the feature vector for every pixel. The size of the pre-processed data is 64×187500. Each column of this matrix represents the feature vector of a pixel, and there are 187500 pixels in the image. Let's denote this matrix as $\boldsymbol{Y}$.

Exercise 1. Naive Bayes

In this part of the exercise, we will build a Naive Bayes classifier. For simplicity we will use a multi-dimensional Gaussian model. To construct the model, we need to know the mean vector and the covariance matrix of each class. These can be computed using `numpy.mean` and `numpy.cov` command. (Note that since this time you will be calculating the mean of each row, you should use `numpy.mean(x,1)`.) If you do everything correct, you will have two vectors $\boldsymbol{\mu}^{(\text{cat})} \in \mathbb{R}^{64\times 1}$ and $\boldsymbol{\mu}^{(\text{grass})} \in \mathbb{R}^{64\times 1}$, and two matrices $\boldsymbol{\Sigma}^{(\text{cat})} \in \mathbb{R}^{64\times 64}$ and $\boldsymbol{\Sigma}^{(\text{grass})} \in \mathbb{R}^{64\times 64}$.

Consider the k-th pixel of the image, or correspondingly $\boldsymbol{y}_k$, the k-th column of the feature matrix $\boldsymbol{Y}$. We will say that this pixel should be **foreground** if

$$\mathcal{N}(\boldsymbol{y}_k \,|\, \boldsymbol{\mu}^{(\text{cat})}, \boldsymbol{\Sigma}^{(\text{cat})})\mathbb{P}[\text{Class Cat}] \geq \mathcal{N}(\boldsymbol{y}_k \,|\, \boldsymbol{\mu}^{(\text{grass})}, \boldsymbol{\Sigma}^{(\text{grass})})\mathbb{P}[\text{Class Grass}], \tag{1}$$

and **background** if

$$\mathcal{N}(\boldsymbol{y}_k \,|\, \boldsymbol{\mu}^{(\text{cat})}, \boldsymbol{\Sigma}^{(\text{cat})})\mathbb{P}[\text{Class Cat}] < \mathcal{N}(\boldsymbol{y}_k \,|\, \boldsymbol{\mu}^{(\text{grass})}, \boldsymbol{\Sigma}^{(\text{grass})})\mathbb{P}[\text{Class Grass}], \tag{2}$$

where

$$\mathcal{N}(\boldsymbol{y}_k \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_k - \boldsymbol{\mu})\right\}.$$

You can take log on both sides of Equation (1) and (2) to bypass the exponential computation.

$$\log\left(\mathcal{N}(\boldsymbol{y}_k \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})\mathbb{P}[\text{Class}]\right) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_k - \boldsymbol{\mu}) + \log(\mathbb{P}[\text{Class}]).$$

In this exercise, we can assume that $\mathbb{P}[\text{Class Grass}] = \mathbb{P}[\text{Class Cat}] = \frac{1}{2}$. Note that $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$ which can be computed by `numpy.linalg.det` command.

Write a for-loop to loop through all the 187500 pixels of the feature matrix $\boldsymbol{Y}$ and decide which pixels should be classified as foreground and which pixel should be classified as background. Mark the foreground as 1 and background as 0. Save your result as a 187500×1 vector `output` and then reshape it to 375×500 using the command `numpy.reshape`. Finally save the image using `matplotlib.pyplot.imsave`.

(Hint: You can pre-compute the inverse $\boldsymbol{\Sigma}^{-1}$ before the for-loop to save some time.)

Exercise 2. KNN
We will now write a KNN classifier for the same dataset. To do so, we pick the k-th column of the $\boldsymbol{Y}$ feature matrix. Call this column vector as $\boldsymbol{y}_k \in \mathbb{R}^{64\times 1}$.

For each class `cat` and `grass`, compute the distance between $\boldsymbol{y}_k$ and the data points in the datasets. That is,

$$
\begin{aligned}
d(\boldsymbol{y}_k, \boldsymbol{x}_j^{(\text{cat})}) &= \|\boldsymbol{y}_k - \boldsymbol{x}_j^{(\text{cat})}\|^2, \\
d(\boldsymbol{y}_k, \boldsymbol{x}_j^{(\text{grass})}) &= \|\boldsymbol{y}_k - \boldsymbol{x}_j^{(\text{grass})}\|^2,
\end{aligned}
$$

where $j = 1, \ldots, 500$. Then, pick the $K = 5$ nearest neighbors by inspecting the distances. (Hint: `numpy.argsort` will return the indices that would sort an array in ascending order.) If there are more samples labeled as `cat`, then return the output as foreground; otherwise as background.

Write a for-loop to loop through all the 187500 pixels of the feature matrix $\boldsymbol{Y}$ and decide which pixels should be classified as foreground and which pixel should be classified as background. Mark the foreground as 1 and background as 0. Save your result as a 187500×1 vector `output` and then reshape it to 375×500 using the command `numpy.reshape`. Finally save the image using `matplotlib.pyplot.imsave`.

(Hint: Looping through all the pixels will take 1-2 minutes.)

