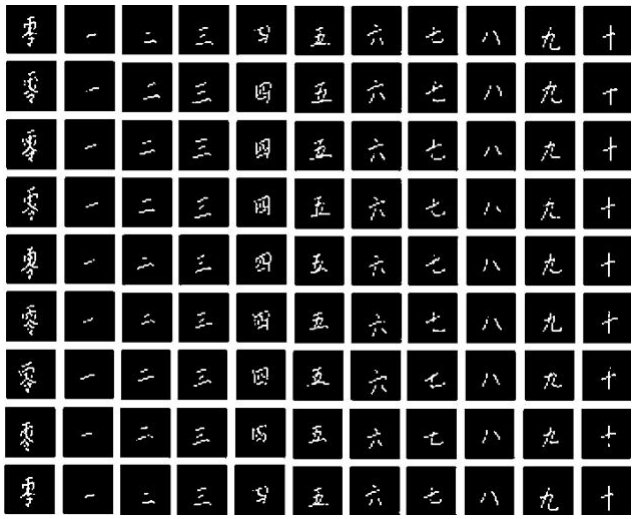
Machine Learning (Homework 3)

Due date : Due date : 2021/12/31 23:59:59

1 Support Vector Machine (SVM) (60%)

Support vector machine (SVM) is known as a popular method for pattern classification. In this exercise, you will implement SVM for classification. Here, the Chinese MNIST dataset is given in x_train.csv and t_train.csv. Chinese MNIST is a dataset collected by the Newcastle University. The input data contain three categories of Chinese numbers: zero, one and four. Each example is a 28x28 gray-scale image, associated with a digit label.



Data Description

- **x_train** is a 300 × 784 matrix, where each row is all pixels of a training image.
- **t_train** is a 300 × 1 matrix, which records the classes of the training images. 0, 1, 2 represent the Chinese numbers: zero, one and four, respectively.

In the training procedure of SVM, we need to optimize with respect to the Lagrange multiplier $\alpha = \{\alpha_n\}$. Here, we use the Sequential Minimal Optimization to solve the problem. For details, you can refer to the paper [Platt, John. "Sequential minimal optimization: A fast algorithm for training support vector machines", 1998]. The classifier is written by

$$y(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n t_n k(\mathbf{x}, \mathbf{x}_n) = \mathbf{w}^\top \mathbf{x} + b$$

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n t_n \phi(\mathbf{x}_n).$$

Scikit-learn is a free software machine learning library which provides sklearn.svm. You are allowed to use the library to calculate the multipliers (coefficients) rather than using the **prediction function** directly. In this exercise, you will implement two kinds of kernel SVM

- **Linear kernel:**

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

- **Polynomial (homogeneous) kernel of degree 2:**

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^2$$

$$\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]$$

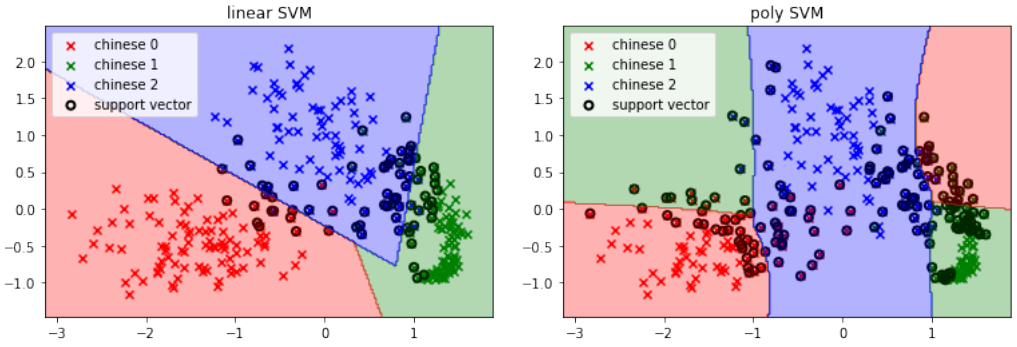
$$\mathbf{x} = [x_1, x_2]$$

SVM is binary classifier, but the application here has three classes. To solve this problem, there are two main decision approaches, one is 'one-versus-the-rest', and another is 'one-versus-one'.

1. Use the principal component analysis (PCA) to reduce the dimension of images to $d = 2$. You can use the sklearn.decomposition.PCA, or implement by yourself to receive the extra bonus 10%.
2. Analyze the difference between two decision approaches (one-versus-the-rest and one-versus-one). Decide which one you want to choose and explain why you choose this approach.
3. Use the principle values projected to top two eigenvectors obtained from PCA, and build a SVM with linear kernel to do multi-class classification. Then, plot the corresponding decision boundary and show the support vector.
4. Repeat (3) with polynomial kernel (degree = 2).
5. Please discuss the difference between (2), (3).

Hints

- You need to implement the whole algorithms except for multipliers (coefficients).



2 Gaussian Mixture Model (40%)

In this exercise, you will implement a Gaussian mixture model (GMM) and apply it in image segmentation. First, use a K-means algorithm to find K central pixels. Second, use Expectation maximization (EM) algorithm (please refer to textbook p.438-p.439) to optimize the parameters of the model. The input data is hw3.jpg. According to the maximum likelihood, you can decide the color μ_k, $k \in [1, \ldots, K]$ of each pixel x_n of output image

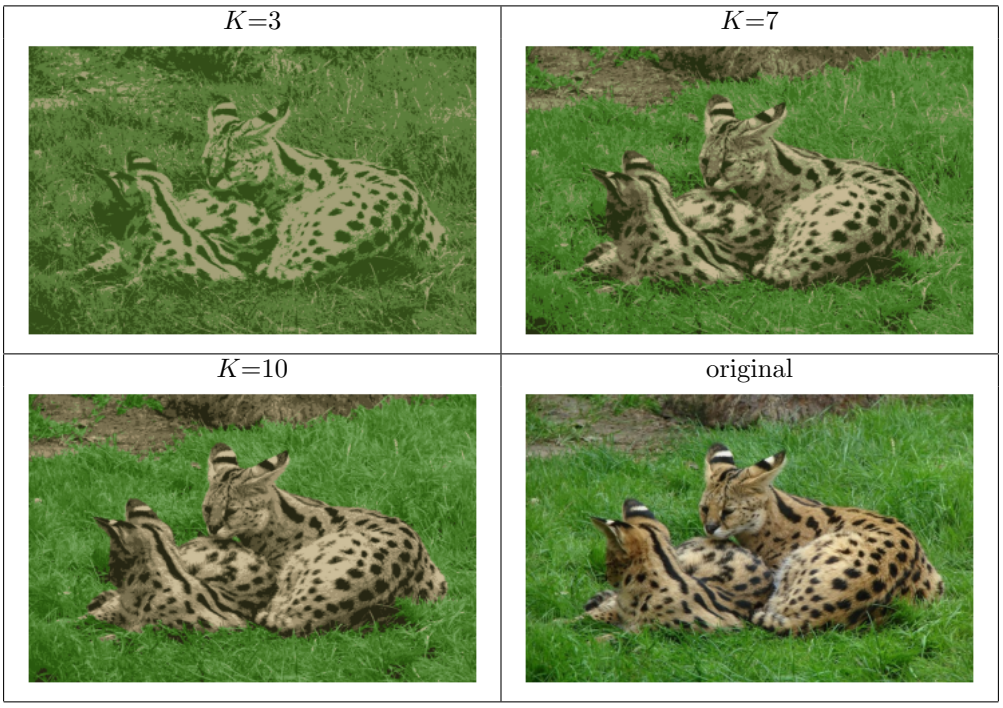
1. Please build a K-means model by minimizing

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \|x_n - \mu_k\|^2$$

   and show the table of the estimated $\{\mu_k\}_{k=1}^{K}$.

2. Use $\{\mu_k\}_{k=1}^{K}$ calculated by the K-means model as the means, and calculate the corresponding variances σ_k^2 and mixing coefficient π_k for the initialization of the GMM $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)$. Optimize the model by maximizing the log likelihood function $\log p(x|\pi, \mu, \sigma^2)$ through EM algorithm. Plot the learning curve for log likelihood of GMM. (Please terminate EM algorithm when the number of iterations arrives at 100.)

3. Repeat steps (1) and (2) for $K = 3$, 7, 10 and 30. Please show the resulting images in your report. Below are some examples.



4. Make some discussion about what is crucial factor to affect the output image between K-means and Gaussian mixture model (GMM), and explain the reason.

5. The input image shown below comes from the licence-free dataset for personal and commercial use. Image from: https://pickupimage.com/free-photos/Red-Panda/2342494

3 Rules

- Please name the assignment as hw3_StudentID.zip (e.g. hw3_0123456.zip).
- In your submission, it needs to contain three files.
  Note : Only the following three files are accepted, so the code of each exercise should be written in one .py file.
  - **.ipynb** file which contains all the results and codes for this homework.
  - **.py** file which is downloaded from the .ipynb file.
  - **.pdf** file which is the report that contains your description for this homework.
- Implementation will be graded by
  - Completeness
  - Algorithm Correctness
  - Model description
  - Discussion
- Only Python implementation is acceptable.
- Only the packages we provided is acceptable.
- DO NOT PLAGIARIZE. (We will check program similarity score.)