# Machine Learning Homework 5

## Gaussian Process & SVM

*==Due Date 23:55 2022/5/12 ==*

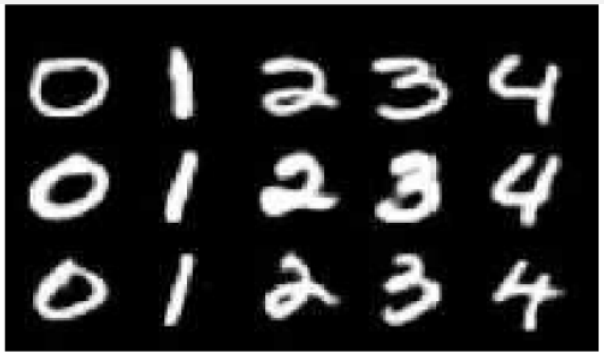
> ⚠️ You are only allowed to use `LIBSVM` library, `numpy`, `scipy.optimize`, `scipy.spatial.distance`, and *package for visualizing results*
> ⚠️ Important: `scikit-learn` is not allowed.

## 1. Gaussian Process

> In this section, you are going to implement the *Gaussian Process* and *Visualize* the result.

- 📄 **Training Data**
  - Given a matrix $A_{34\times 2}$, representing $34$ data entries, each row corresponds to a 2D data point $(x_i, y_i)$.
    - You have to read the matrix $A$ from `data/input.data` in hw5 zip file.
  - Let's assume that $A$ represents a *noisy observation of an unknown function* $f$, in particular, $y_i = f(x_i) + \epsilon$ , where $\epsilon \sim N(\cdot\,|0,\ \beta^{-1})$.
    - Please use $\beta$ = 5 in this implementation.
- 🎯 **Task 1**
  - 👍Apply Gaussian Process Regression to predict the distribution of $f$
    - Please use a *rational quadratic kernel* to compute similarities between different points
  - 👍Visualize the prediction result .
    - Show all training data points.
    - Draw a line to represent mean of $f$ in range $[-60, 60]$
    - Mark the 95% confidence interval of $f$
    - You can use `matplotlib.pyplot.fill_between` to show the confidence interval, or *any other way you like*.
- 🎯 **Task2:**
  - 👍Optimize the *Kernel Parameters* by minimizing *negative marginal log-likelihood*
    - You can use `scipy.optimize.minimize` to optimize the parameters, or *any other way you like*.
  - 👍Visualize the result again.

## 2. SVM on MNIST

> Use SVM models to tackle classification on images of hand-written digits (digit class only ranges from 0 to 4, as the figure shown below).
> 

- 📄 **Training data**
  - `data/X_train.csv` is a 5000×784 matrix. Every row corresponds to a 28×28 gray-scale image.
  - `data/Y_train.csv` is a 5000×1 matrix, which records the class of the training samples.
- 📄 **Testing data**
  - `X_test.csv` is a 2500×784 matrix. Every row corresponds to a 28×28 gray-scale image.
  - `Y_test.cs` is a 2500×1 matrix, which records the class of the test samples.
- 🎯**Task1:**

    - Use different kernel functions (linear, polynomial, and RBF kernels) and have comparison between their performance.
- 🎯Task2:
    - Please use C-SVC (you can choose by setting parameters in the function input, C-SVC is soft-margin SVM).
    - Since there are some parameters you need to tune for, please do the grid search for finding parameters of the best performing model.
        - For instance, in C-SVC you have a parameter $C$, if you use RBF kernel you have another parameter $\gamma$. Now you can search for a set of $(C, \gamma)$ which gives you best performance in cross-validation. ( There are lots of sources on the internet, just google for it )
- 🎯Task3:
    - Use **linear kernel + RBF kernel** together (therefore a new kernel function) and compare its performance with respect to others.
        - You would need to find out how to use a *user-defined kernel* in `libsvm`.

## 3. Submitting Report

### Important Rules

- Submit a report in *pdf format*. The report should be written in *English*.
- Please follow the **report format**. If you skip some sections in the report format, you score will be affected. Additional content outside the format is welcome ( but we may not be able to give you extra points ).
- Please don't explain the code line by line. You need to explain it clearly and well structured. For example, explain which part you done in the function.
- Since this homework is mainly graded by report, please spend more time on it. (e.g. well organized) We won't give you any points if you just finish the code.

### Report Format

- **I. Gaussian Process**
    1. **Code (20%) :**
        - Expected Content:
            - Show code with detailed explanations
            - For example, show the formula of rational quadratic kernel and the process you optimize the kernel parameters
            - ⚠️ *Note that if you don't explain your code, you cannot get any points in section 2 and 3 either.*
        - Code for Task 1 (10%)
        - Code for Task 2 (10%)
    2. **Experiments(20%)**
        - Expected Content:
            - Show experiment Settings and Results, including the figures and the hyperparameters we asked you to show.
            - Note that if you don't explain your code in the above section, you cannot get any points in this section either.
        - Experiment for Task 1 (10%)
        - Experiment for Task 2 (10%)
    3. **Observations and Discussion (10%)**
        - Anything you want to discuss, such as comparing the performance when using different hyperparameters.
        - If you need to refere to images or code snippets in previous sections, you can either
            i. Add a duplicate one in this section
            ii. Add title or numbering system to all images, and refer to the image by thier corresponding identifier.
            iii. Put some parts of your discussions in the middle of previous sections, but you have to make it super obvious for us. ( add title or headings to tell us that the paragraph is part of "Observations and Discussion").
- **II. SVM on MNIST**
    1. **Code (20%)**
        - Expected Content
            - Paste the screenshot of your functions with comments and explain your code.
            - For example, show the formula of different kernel functions and the process you search for the kernel parameters, etc.
            - ⚠️*Note that if you don't explain your code, you cannot get any points in section 2 and 3 either.*
        - Code for Task1 (5%)
        - Code for Task2 (10%)
        - Code for Task3 (5%)
    2. **Experiments (20%)**

    - Expected Content
        - Show experiment Settings and Results, including everythings we asked you to show.
    - Experiment for Task1 (6%)
    - Experiment for Task2 (8%)
    - Experiment for Task3 (6%)
3. **Observations and Discussion (10%)**
    - Anything you want to discuss, such as trying different user-defined kernel functions and comparing the performance.
    - If you need to refere to images or code snippets in previous sections, you can either
        i. Add a duplicate one in this section
        ii. Add title or numbering system to all images, and refer to the image by thier corresponding identifier.
        iii. Put some parts of your discussion in the middle of previous sections, but you have to make it super obvious for us. ( add title or headings to tell us that the paragraph is part of "Observations and Discussion").

### Alternative Ordering

★ You may also re-arrange the order of each section in the following way, our grading rules will be identical.

- Gaussian Process
    - Task1
        - Code
        - Experiment
        - Discussion
    - Task2
        - Code
        - Experiment
        - Discussion
    - Conclusion (Optional)
- SVM on MNIST
    - Task1
        - Code
        - Experiment
        - Discussion
    - Task2
        - Code
        - Experiment
        - Discussion
    - Task3
        - Code
        - Experiment
        - Discussion
    - Conclusion (Optional)

## 4. Submision

To Submit the Homework, you should:

- Zip your contents in one file, including
    - Report (.pdf)
    - Source Code
- Name the zip file as `ML_HW5_{your student id}_{your name}.zip`. (e.g. ML*HW5_0856XXX*王小明.zip)
    - If the zip file name has format error or the report is not in pdf format, there will be a penalty (we are considering -10, please be scared).
- Submit your homework *in time*
    - After deadline, you can still sumit in the following 7 days, you will get only 70% of original score.
    - Starting from the seven'th day after the deadline, you can not submit your homework and you will get 0 score.
    - Whenever you submit your homework, the latest submission will be used for grading. (so don't accidently submit something after deadline, you will get 70% discount no matter what)