Machine Learning Homework 5

Gaussian Process & SVM

Due Date 23:55 2022/12/04

I. Gaussian Process

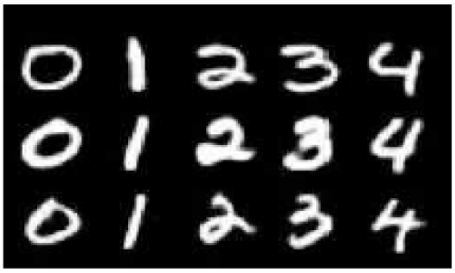
In this section, you are going to implement the Gaussian Process and visualize the result.

- Training data
  - **input.data** is a 34x2 matrix. Every row corresponds to a 2D data point (X_i,Y_i).
  - $Y_i = f(X_i) + \epsilon_i$ is a noisy observation, where $\epsilon_i \sim N(\bullet|0, \beta^{-1})$. You can use $\beta = 5$ in this implementation.
- What you are going to do
  - Part1: Apply Gaussian Process Regression to predict the distribution of f and visualize the result. Please use a rational quadratic kernel to compute similarities between different points.
    Details of the visualization:
    - Show all training data points.
    - Draw a line to represent the mean of f in range [-60,60].
    - Mark the 95% confidence interval of f.

    (You can use matplotlib.pyplot to visualize the result, e.g. use matplotlib.pyplot.fill_between to mark the 95% confidence interval, or you can use any other package you like.)
  - Part2: Optimize the kernel parameters by minimizing negative marginal log-likelihood, and visualize the result again. (You can use scipy.optimize.minimize to optimize the parameters.)

II. SVM on MNIST dataset

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

    - **X_train.csv** is a 5000x784 matrix. Every row corresponds to a 28x28 gray-scale image.
    - **Y_train.csv** is a 5000x1 matrix, which records the class of the training samples.
- Testing data
    - **X_test.csv** is a 2500x784 matrix. Every row corresponds to a 28x28 gray-scale image.
    - **Y_test.csv** is a 2500x1 matrix, which records the class of the test samples.
- What you are going to do
    - Part1: Use different kernel functions (linear, polynomial, and RBF kernels) and compare their performance.
    - Part2: 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, and if you use RBF kernel you have another parameter γ, you can search for a set of (C, γ) which gives you best performance in cross-validation. (There are lots of sources on the internet, just google for it.)
    - Part3: Use linear kernel + RBF kernel together (therefore a new kernel function) and use grid search again. You would need to find out how to use a user-defined kernel in libsvm.

III. Report

- Submit a report in pdf format. The report should be written in English.
- Please strictly follow the report format. We will deduct some points according to the situation if you don't follow it.
- Please don't explain the code line by line. You need to explain it clearly and well structurally. For example, explain what the function is used for and explain what formula you have used 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
        - a. code with detailed explanations (20%)
            - 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.

        - Part1 (10%)
        - Part2 (10%)
    - b. experiments settings and results (20%)
        - Show 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.
            - Part1 (10%)
            - Part2 (10%)
    - c. observations and discussion (10%)
        - Compare the performance when using different hyperparameters.
        - Anything you want to discuss.

- II. SVM
    - a. code with detailed explanations (20%)
        - 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 b and c either.
            - Part1 (5%)
            - Part2 (9%)
            - Part3 (6%)
    - b. experiments settings and results (20%)
        - Show everything we asked you to show
            - Part1 (6%)
            - Part2 (8%)
            - Part3 (6%)
    - c. observations and discussion (10%)
        - Explain why some kernels have better performance than others.
        - Try different user-defined kernel functions and compare the performance.
        - Anything you want to discuss.

IV. Turn in

1. Report (.pdf)
2. Source code

You should compress your source code and the report into a zip file and name it like ML_HW5_yourstudentID_name.zip, e.g. ML_HW5_0856XXX_王小明.zip.

P.S. If the zip file name has format error or the report is not in pdf format, there will be a penalty (-10). Please submit your homework before the deadline. After the deadline, you can still submit your homework in the following 7 days, but you will only get 70% of the original score.

- Packages allowed in this assignment:

  You are only allowed to use the LIBSVM library, numpy, scipy.optimize, scipy.spatial.distance, and package for visualizing results. Official introductions can be found online.
  Important: scikit-learn is not allowed.