Machine Learning Homework 5

Gaussian Process and SVM

Due Date 23:55 31th May.

I. Gaussian Process (35% in total)

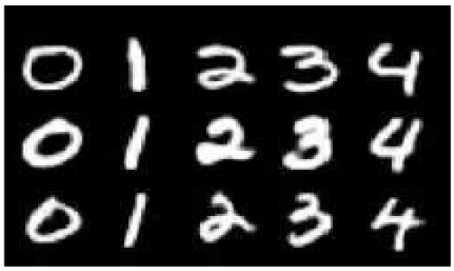
In this section, you are going to implement Gaussian Process and visualize the result.

- 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 $\epsilon_i \sim N(\cdot|0, \beta^{-1})$. You can use $\beta = 5$ in this implementation.
- What you are going to do
  - (20%) Apply Gaussian Process Regression to predict the distribution of *f* and visualize the result. Please use rational quadratic kernel to compute similarities between different points.
    Details of the visualization:
    - 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 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.)
  - (15%) 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 (45% in total)

Use SVM models to tackle classification on images of hand-written digits (digit class only ranges from 0 to 4, as 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
  - (10%) Use different kernel functions (linear, polynomial, and RBF kernels) and have comparison between their performance.
  - (20%) 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 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 internet, just google for it.)
  - (15%) 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.

III. Report (20% in total)

Submit a report in pdf format for showing your code with detailed explanations, giving detailed discussion on experiments as well as your observations. The report should be written in English.

Noted that if you don't explain your code in the report, you cannot get points at I and II section.

IV. Turn in

1. Report (.pdf)
2. Source code

You should zip source code and report in one 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 deadline, late submission is not allowed.

♦ Packages allowed in this assignment:

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