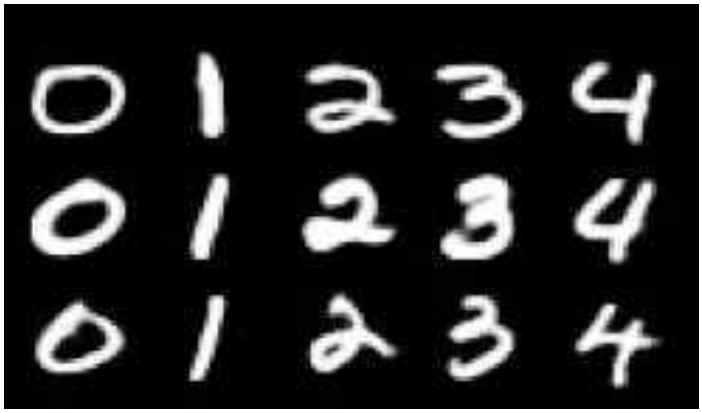
Machine Learning 2017

HW-4：Supporting Vector Machine

Deadlines: **2017.05.2**-23:59:59

In this homework, you are asked to use **SVM** models dealing with the classification problem of the hand-written numbers from zero to four. The whole mission can be accomplished easily with the **LIBSVM** library which is highly recommended to adopt. The official website of LIBSVM is shown bellowed:

- https://www.csie.ntu.edu.tw/~cjlin/libsvm/#download



◆ Data

Training _data

- **X_train.csv** is a 5000x784 matrix. Every row corresponds to a 28x28 gray-scale image.
- **T_train.csv** is a 5000x1 matrix, which records the class of the training samples.

Test_data

- **X_test.csv** is a 2500x784 matrix. Every row corresponds to a 28x28 gray-scale image
- **T_test.csv** is a 2500x1 matrix, which records the class of the test samples.

(Please do remember that the test data is not used for optimizing your model!!)

◆ Installation Guide

● Matlab users

1. The **Visual Studio** is necessary for compiling mex files. Please install the corresponding Visual Studio version by yourself.

2. Download the LIBSVM package from the website and unzip it:
   http://www.csie.ntu.edu.tw/~cjlin/cgi-bin/libsvm.cgi?+http://www.csie.ntu.edu.tw/~cjlin/libsvm+zip

3. Open the **Matlab** software and set the path to /path_to_libsvm-3.21/matlab/ and type the commands bellow:

```
matlab>> mex -setup
matlab>> make
```

4. Read the **README** file in the folder and try to implement the LIBSVM models. The most important two code lines would be:

```
model = svmtrain(T_train, X_train, ['options']);
[predict_label, accuracy, prob_estimates] =...
    svmpredict(T_test, X_test, model, '-b 1');
```

- **Python & C++ users**

  Please help yourself to find the information on the LIBSVM website. https://www.csie.ntu.edu.tw/~cjlin/libsvm/#download

About the SVM **training options**, please take a look at the webpage of LIBSVM. If you are planning for **changing the kernel**, you should do something to the training options!

◆ Models

- **C-SVM**

According to the lecture note, the problem of SVM can be transferred into the dual representation:

$$\text{maximize} \quad \tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

$$\text{subject to} \quad 0 \le a_n \le C$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

- **ν-SVM**

An alternative, equivalent formulation of the SVM:

$$\text{maximize} \quad \tilde{L}(a) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

$$\text{subject to} \quad 0 \le a_n \le 1/N$$

$$\sum_{n=1}^{N} a_n t_n = 0 \quad \text{and} \quad \sum_{n=1}^{N} a_n \ge \nu$$

◆ Tasks

1. ν-SVM

Implement the ν-SVM models with the different kernel types below

- **Linear function**
- **Polynomial function** with degree = 2,3,4
- **Radial basis function**

And compare the performance between them.

2. C-SVM

Implement the ν-SVM models with the different kernel types below

- **Linear function**
- **Polynomial function** with degree = 2,3,4
- **Radial basis function**

And compare the performance between them. **If you design the ν properly, you might get the same outcomes with Task 1.**

3. Supporting Vectors

Choose one of the models you've trained to do analysis and find those "**supporting vectors**" and "**outliners**" for this model. You should use PCA to map data down to two dimensions

Please plot at least 30 samples for each and discuss your observations.

What should be uploaded?

- ☐ Your source code with comments.
- ☐ The **ReadMe.txt** file which describes how to run your program.
- ☐ Your **report** in the format of .pdf or .doc.

Reminders:

- ☐ Report within 12 pages
- ☐ There won't be a need for demonstration.
- ☐ Please make sure your source code can be compiled by **Matlab, Python or C++**.
- ☐ **DO NOT COPY!!!**（懶人包、考古題亦同）