Multimedia Cloud Computing and Machine Learning

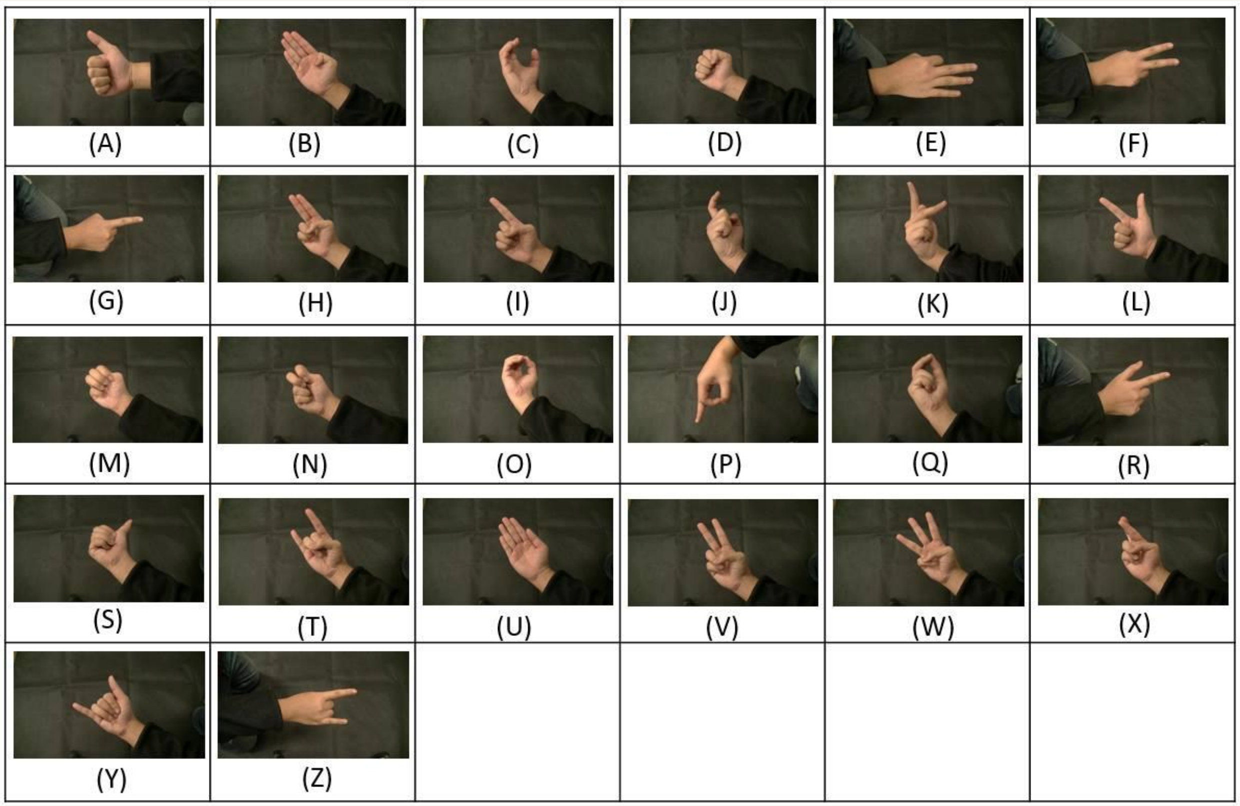
Lecturer: Min-Chun Hu 胡敏君

Homework 2 : SVM

Deadline: 11/5 pm 10:00

SVM Library

1. Extract features to describe the hand gesture and apply SVM to classify the “CSL” dataset. (50%)



SVM Library

- Dataset (https://goo.gl/12JsfT)
- The submitted file should contain
  1. Source code
  2. A report including:
     - Brief description of development environment
     - Description of your features
     - Comparison of
       - the effect of scaling the feature values
       - using different SVM kernels and parameters
     - The cross validation results
       - the precision/recall of each validation iteration
       - the average precision/recall of all validation iterations
     - Description of your observations

Implement SVM

2. Implement Support Vector Machine.

- **Implement SMO (without kernel) (30%)**
- **Implement Heuristic Choosing Algorithm (20%)**
- **Implement RBF kernel (10%)**

Dataset: *liver-disorders_scale.txt*

```
0 1:0.904762 2:-0.391304 3:-0.505376 4:-0.384615 5:-0.767677
0 1:0.52381 2:-0.269565 3:-0.397849 4:-0.153846 5:-0.878788
0 1:0.238095 2:-0.165217 3:-0.956989 4:-0.346154 5:-0.939394
0 1:0.142857 2:-0.443478 3:-0.784946 4:-0.423077 5:-0.888889
1 1:0.238095 2:-0.46087 3:-0.677419 4:-0.346154 5:-0.69697
1 1:-0.142857 2:-0.46087 3:-0.333333 4:-0.0769231 5:-0.818182
1 1:0.333333 2:-0.234783 3:-0.892473 4:-0.653846 5:-0.909091
1 1:0.904762 2:0.356522 3:-0.548387 4:-0.192308 5:-0.727273
```

Implement SVM

Sequential Minimal Optimization:

A Fast Algorithm for Training Support Vector Machines

John C. Platt
Microsoft Research
jplatt@microsoft.com
Technical Report MSR-TR-98-14
April 21, 1998

ABSTRACT

This paper proposes a new algorithm for training support vector machines: *Sequential Minimal Optimization,* or *SMO*. Training a support vector machine requires the solution of a very large quadratic programming (QP) optimization problem. SMO breaks this large QP problem into a series of smallest possible QP problems. These small QP problems are solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets. Because matrix computation is avoided, SMO scales somewhere between linear and quadratic in the training set size for various test problems, while the standard chunking SVM algorithm scales somewhere between linear and cubic in the training set size. SMO's computation time is dominated by SVM evaluation, hence SMO is fastest for linear SVMs and sparse data sets. On real-world sparse data sets, SMO can be more than 1000 times faster than the chunking algorithm.

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-98-14.pdf

Implement SVM

- Dataset (https://goo.gl/12JsfT)
- The submitted file should contain
  1. Source code
  2. A report including:
     - Brief description of development environment
     - Result of α, $\mathbf{Accuracy}$ of the given $\mathbf{C}$, σ
     - Description of your algorithm

Joint Optimization of Multipliers

3. Proof the jointly optimization result (20%):

$$\alpha_2^{new} = \alpha_2^{old} - \frac{y^{(2)}(E_1 - E_2)}{\eta}$$

***Hint*:**

The error on the i-th training sample : $E_i = \left(\mathbf{w}^{old} \cdot \mathbf{x}^{(i)} + b\right) - y^{(i)}$

$$\alpha_1 = \gamma - s\alpha_2$$