Neural Network Theory and Applications
Homework Assignment 3

April 7, 2015
Due at April 19, 2015

This homework requires you to implement the multi-class SVM classification based on the LibSVM package, which can be freely downloaded from http://www.csie.ntu.edu.tw/∼cjlin/libsvm/. Alternatively, you can choose SVM^{light}.

Requirement

1. Implement traditional one-versus-one and one-versus-rest task decomposition methods to solve the multi-class problem.
2. Implement part-versus-part task decomposition method.
3. Compare the advantages and disadvantages of these three task decomposition methods.
4. Try three different kernel functions, namely linear, polynomial and RBF, and make a comparison on training time and generalization performance.

Dataset

The dataset (train.txt, test.txt) contains protein sequences from 12 subcellular locations. 20 dimensions stand for 20 amino acid composition of the protein sequences.

No. of proteins: 7579 (6065 training samples, 1514 test samples).

No. of classes: 12 ($0 \sim 11$).

The data file format:

label	dim1 : value	dim2 : value	$\cdots$	dim20 : value
0	1 : 0.095861	2 : 0.010893	$\cdots$	20 : 0.032680

Description of LibSVM

Since SVM is so popular in nowadays machine learning research, I believe you will benefit a lot from this homework.

In this homework, you should only use two files of LibSVM, svm.h and svm.cpp. And three interface functions listed below:

1. **struct svm_model *svm_train(const struct svm_problem *prob, const struct svm_parameter *param)**;
2. **void svm_predict_values(const struct svm_model *model, const struct svm_node *x, double* dec_values);**
   Call this function to get the decision values of prediction, instead of class labels. You need to call it to construct one-versus-others and part-versus-part classifier.
3. **double svm_predict(const struct svm_model *model, const struct svm_node *x);**
   Call this function to get the class labels of prediction. It worth mentioning that LibSVM has implemented one-versus-one multi-class classifiers. However, in this homework, you should construct one-versus-one from only binary classifier, without directly use the code of LibSVM.

To call these three functions, you should also be familiar with four structs, svm_problem, svm_model, svm_parameter and struct svm_node, all of which can be found in svm.h and svm.cpp.

In summary, although the length of LibSVM source code is very huge, you can crack it by using these functions above. Surely you can make use of other language versions of C#, java, Matlab and so on.