**Liblinear one-vs-all-classifier-implementation**

%# Fisher Iris dataset

load fisheriris

[~,~,labels] = unique(species); %# labels: 1/2/3

data = zscore(meas); %# scale features

numInst = size(data,1);

numLabels = max(labels);

%# split training/testing

idx = randperm(numInst);

numTrain = 100; numTest = numInst - numTrain;

trainData = data(idx(1:numTrain),:); testData = data(idx(numTrain+1:end),:);

trainLabel = labels(idx(1:numTrain)); testLabel = labels(idx(numTrain+1:end));

Here is my implementation for the one-against-all approach for multi-class SVM:

%# train one-against-all models

model = train(double(trainLabel), sparse(trainData), '-c 1 -B 1')

[pred, accuracyVal ,prob] = predict(double(testLabel), sparse(testData), model, '-b 1');

acc = sum(pred == testLabel) ./ numel(testLabel) %# accuracy

C = confusionmat(testLabel, pred) %# confusion matrix

FAQ: http://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html#/Q9:\_MATLAB\_interface

Another method>> Use ovrtrain and ovrpredict wrapper