Homework 4 – Machine Learning

Saeid Hosseinipoor

Problem 1 – SVM Weight Vector

Minimizing the norm of the weight vector w which is minimizing $J(w)=\frac{1}{2}\|w\|^2$, is equivalent to maximizing $m=\frac{2}{\|w\|}$which is the margin between closest samples and hyperplane. It means that we have bigger space between objects close to the support vector (hyperplane) which classifies different classes and we will have more accurate classification.

Problem 2 – LibLinear

The library has been installed and used to solve the problem with arguments '-c 1 -v 5 -q' for C=1 and 5fold in quiet mode. The code is attached. I ran the algorithm 10 times and shuffled the data for each run. Average accuracy for 10 runs was around 56%.

Problem 3 – AdaBoost

I have followed the procedure and calculated the weak learners, their directions, strong learners, their directions and accuracies. The code is attached. This the sample output:

```
The accuracy for round 1 is 75.00% with following weak learners:
   6.0000  143.0000   96.0000   43.0000  142.0000   40.8000    1.1010   40.0000
The accuracy for round 2 is 75.00% with following weak learners:
   6.0000  106.0000   78.0000   31.0000  142.0000   29.9000    0.5010   28.0000
The accuracy for round 3 is 75.00% with following weak learners:
   6.0000  111.0000   44.0000   31.0000  142.0000   29.9000    0.5010   21.0000
Strong features are:
    2     8     6
The final accuracy is 75.00% with following strong learners:
  143.0000   28.0000   29.9000
The directions of strong selections are:
    1     1     1
Note:
             1 means that greater than theta is in class 1 and,
             2 means that greater than theta is in class 2.
```

Appendix 1 – MATLAB Code

```
clear

close all
clc

tic

data = dlmread('pima-indians-diabetes.data.txt');
data = reshape(data,[],9);
active_feat = 2:4;

accuracy = zeros(1,10);
for i = 1:10
    rp = randperm(length(data));
    data=data(rp,:);
    train_data = sparse(data(:, active_feat));
    train_label = data(:,end);
    model = train(train_label, train_data,'-c 1 -v 5 -q');
    accuracy(i) = model;
end

fprintf('\n\nThe mean accuracy is %4.2f%%:\n\n\n', mean(accuracy))

toc
```

```
clear

close all
clc

tic

AdaRun = 3;


data = dlmread('pima-indians-diabetes.data.txt');
data = reshape(data,[],9);
active_feat = 1:8;

X = data(:,active_feat);
Y = data(:,end);
Y(Y == 0) = -1;

[N, f] = size(X);

D = 1 / N * ones(N,1);

alpha = zeros(1,AdaRun);
h = zeros(N,AdaRun);
```

```
strong_features = zeros(1,AdaRun);
strong_learners = zeros(1,AdaRun);
strong_directions = zeros(1,AdaRun);

weak_learners = zeros(f,AdaRun);
weak_directions = zeros(f,AdaRun);

for r = 1:AdaRun

    min_error = inf;

    for i = 1:f
        theta = unique(X(:,i));
        weak_error = inf;

        for j = 1:numel(theta)

            % Positive direction
            pred_labels = 2 * (X(:,i) > theta(j)) - 1;
            err = sum((Y ~= pred_labels) .* D) ./ sum(D);
            if err < weak_error
                weak_learners(i,r) = theta(j);
                weak_directions(i,r) = 1;
                weak_error = err;
            end
            if err < min_error
                strong_features(r) = i;
                min_error = err;
                strong_directions(r) = 1;
                h(:,r) = pred_labels;
                strong_learners(r) = theta(j);
            end

            % Negative Direction
            pred_labels = 2 * (X(:,i) < theta(j)) - 1;
            err = sum((Y ~= pred_labels) .* D) ./ sum(D);
            if err < weak_error
                weak_learners(i,r) = theta(j);
                weak_directions(i,r) = 2;
                weak_error = err;
            end
            if err < min_error
                strong_features(r) = i;
                min_error = err;
                strong_directions(r) = 2;
                h(:,r) = pred_labels;
                strong_learners(r) = theta(j);
            end

        end
    end

    alpha(r) = 0.5 * log((1-min_error)/min_error);
    D = D .* exp(-alpha(r) .* Y .* h(:,r)) ...
        ./ sum(exp(-alpha(r) .* Y .* h(:,r)));
    H = sign(h(:,1:r) * alpha(1,1:r)');
    fprintf(['\nThe accuracy for round %d is %4.2f%% ' ...
        'with following weak learners:\n\n'], ...
        r, sum(H==Y)/N*100)
    disp(weak_learners(:,r)')

end

H = sign(h * alpha');
fprintf ('\n\n\nStrong features are:\n\n')
disp(strong_features)
fprintf(['The final accuracy is %4.2f%% with following ' ...
    'strong learners:\n\n'], sum(H==Y)/N*100)
disp(strong_learners)
fprintf ('The directions of strong selections are:\n')
disp(strong_directions)
fprintf (['Note: \n\t1 means that greater than theat is in class 1 and, ' ...
        '\n\t2 means that greater than theta is in class 2.\n\n\n'])



toc
```