

## Adaptive Boosting

1. train (訓練資料), train label (訓練資料的預期值), cycle (循環次數，產生 weaklearner 次數), step (每次 iterate 所增加的資料筆數), test(測試資料), test label (測試資料的預期值)
2. 可以從 weakLearner.m 觀察到以下程式碼，計算出 error 之後取與 0.5(亂猜) 最遙遠者。演算法即 weakLearner.m。不是 bootstrap，都是取原本 dataset 的原始資料且沒有重複選取來當作同一個小 dataset，如下圖：

```
function [i,t] = weakLearner(distribution,train,label)
    %disp('run weakLearner');
    for tt=1:(16*256-1)
        error(tt)=distribution*abs(label-(train(:,floor(tt/16)+1)>=16*(mod(tt,16)+1)));
    end
    [val,tt]=max(abs(error-0.5));

    i=floor(tt/16)+1;
    t=16*(mod(tt,16)+1);
```

```
function runAdaBoosting(train,train_label,test,test_label)
    disp('run adaboost with cycles=100');
    step=100;
    cycles=100;
    for m=step:step:1000
        disp(m);
        boost=adaBoost(train(1:m,:),train_label(1:m),cycles);
        [errorTrain(m/step),errorTest(m/step)]=getError(boost,train(1:m,:),train_label(1:m),test,test_label);
    end
```

3. s 都是 1 因為程式碼中的  $\text{train}(:, i) \geq t$ ，3 次的 i(feature 編號) 與 theta (樣本邊界) 如下，取 index 2, 3，附上 source code 與實作 code：

```
label=(train_label(:)>=5);% contain the correct label per vector
for j=1:cycles
    if(mod(j,10)==0)
        disp([j,cycles]);
    end
    [i,t]=weakLearner(distribution,train,label);
    error(j)=distribution*abs(label- train(:,i)>=t));
    beta(j)=error(j)/(1-error(j));
    boosted(j,:)=[beta(j),i,t];
    distribution=distribution.* exp(log(beta(j))*(1-abs(label-(train(:,i)>=t)))));
    distribution=distribution/sum(distribution);
end
```

```
fprintf("-----> %d %d\n", boost(1, 2), boost(1, 3))
fprintf("-----> %d %d\n", boost(2, 2), boost(2, 3))
fprintf("-----> %d %d\n", boost(3, 2), boost(3, 3))
```

```
-----> 11 80
-----> 170 80
-----> 58 16
-----> 2 774105
```

4. 實作結果，附上 source code 與實作 code，beta(j) 應該就是 blending weight，故我取 index 1：

```

        if(mod(j,10)==0)
            disp([j,cycles]);
        end
        [i,t]=weakLearner(distribution,train,label);
        error(j)=distribution*abs(label-(train(:,i)>=t));
        beta(j)=error(j)/(1-error(j));
        boosted(j,:)=[beta(j),i,t];
        distribution=distribution.* exp(log(beta(j))*(1-abs(label-(train(:,i)>=t))))';
        distribution=distribution/sum(distribution);
    end
end

```

```
fprintf("----->  %d %d %d\n", boost(1, 1), boost(2, 1), boost(3, 1))
```

```
----->  3.774105e-01 4.806732e-01 2.016979e+00
```