資工系100062236林修安

Machine Learning Assignment 2 Report

- Implementation

+ `phi(X, lift)`: Lift the input instances to feature space, where `lift` denotes type of lifting function.

$$\Phi_1(X)=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \Phi_2(X)=\begin{bmatrix} x_1^2 & x_1 \\ x_2^2 & x_2 \\ \vdots & \vdots \\ x_N^2 & x_N \end{bmatrix} \quad \Phi_3(X)=\begin{bmatrix} x_1^3 & x_1^2 & x_1 \\ x_2^3 & x_2^2 & x_2 \\ \vdots & \vdots & \vdots \\ x_N^3 & x_N^2 & x_N \end{bmatrix}$$

```
function featureX = phi(X, lift)
    featureX = X;
    while lift>1
        featureX = [X.^lift featureX];
        lift = lift-1;
    end
end
```

+ `train(X, r, lift)`:

- Initial W, B randomly in normal distribution.
- Set error tolerance ϵ =10^{-10}
- Iterations from 0 to 10000

For every iteration,

- Set $\eta=\frac{\|(r-y)\Phi(X)\|}{3}$
- Implement

$$W^{(i+1)}=W^{(i)}+\eta\sum_{t=1}^{N}(r^{(t)}-y^{(t)})\Phi(x^{(t)})$$

$$B^{(i+1)}=B^{(i)}+\eta\sum_{t=1}^{N}(r^{(t)}-y^{(t)})$$

- Halt if the convergence $\|W^{(i+1)}-W^{(i)}\|<\epsilon$ occurs, or the maximum iteration is reached.

```
function perceptronClassifierObj = train(X, r, lift)
    featureX = PerceptronClassifier.phi(X, lift);
    W = randn(lift, 1);
    B = randn;
    err = 1e-10;
    for i=0:10000
        preW = W;
        preB = B;
        predictY = featureX*W + B;
        y = sign(predictY);
        eta = norm((r-y)'*featureX)/3;
        W = preW + eta*((r-y)'*featureX)';
        B = preB + eta*sum(r-y);
        if norm(W-preW)<err
            break;
        end
    end
    perceptronClassifierObj = PerceptronClassifier(W, B);
end
```

+ `Generalization(X, r, lift)`:

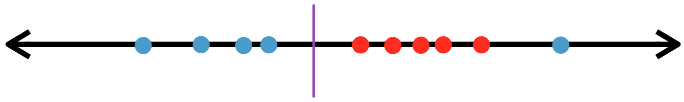
Generalization error is defined as the average of number of incorrect prediction.

```
function generalErr = GeneralizationError(obj, X, r, N, lift)
    liftX = obj.phi(X, lift);
    predictY = liftX*obj.w + obj.b;
    y = sign(predictY);
    generalErr = sum(y~=r)/N;
end
```

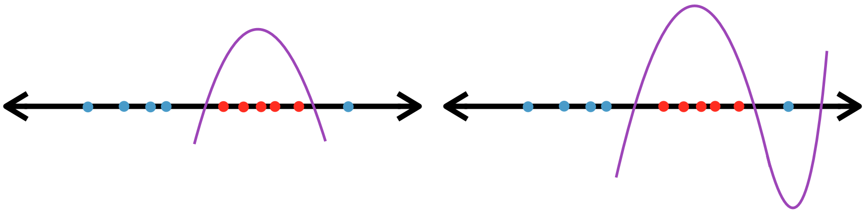
資工系100062236林修安

+ `ConsistencyBound(N, lift)`:

For $\phi_1(X)$, the optimal way to shatter instances by a line is depicted as below. Since 10% of the instances will be mispredicted, the R[h*] = N/10.



For $\phi_2(X)$ and $\phi_3(X)$, it is always able to shatter instances, so the R[h*] = 0;



```
function bound = ConsistencyBound(obj, N, lift)
    if(lift==10), Rh = 1/10;
    else          Rh = 0;
    end
    vc = obj.VCDimension(lift);
    bound = Rh+2*sqrt((32/N)*(vc*log10(N*exp(1)/vc)+log10(40)));
end

function vc = VCDimension(~, lift)
    vc = lift+1;
end
```

- Results

As N increases:

(1) generalization error decreases because we have more history data to train and obtain a more accurate model.

(2) consistency bound decreases. Observed from the formula of the bound, we can tighten the bound with larger N.

As dimension of feature space increases:

(1) generalization error decreases. Because higher dimension model is more likely to shatter instances, thus prediction becomes more accurate.

(2) consistency bound increases because the VC dimension increases.