Machine Learning End Term Solutions Fall 2017

D. Nithish - IMT2014016
R. Vineet Reddy - IMT2014045
V. Keerthi Chandra - IMT2014064

Question 1

The implementation of the Feature Weighted Linear Stacking algorithm can be found in the file **fwls.py**. To execute the file run `make` all or just `make`. To know how to run the code click on the below image 1 or click here.

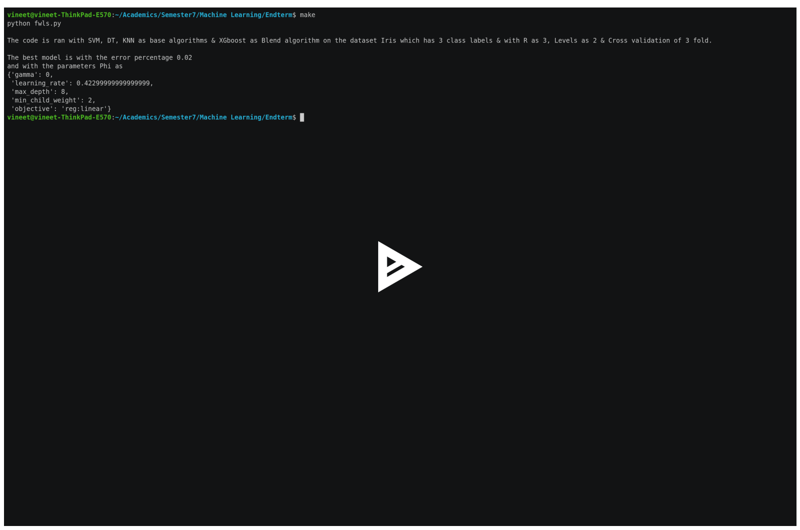

Figure 1: Snippet of how to run the code.

Question 2

Let ϵ be the error matrix of dimensions of n rows & d+1 columns to make it homogeneous, where every ϵ_i is I.I.D and has co-variance matrix σ^2I.

$$W^* = \underset{W}{arg_{min}}\ E_\epsilon[(Y-(X+\epsilon)W)^2]$$

Simplifying the expectation term with respect to attribute noise

$$\begin{aligned}
&= E_\epsilon[(Y-(X+\epsilon)W)^T(Y-(X+\epsilon)W)] \\
&= E_\epsilon[(Y^T - W^TX^T - W^T\epsilon^T)(Y - XW - \epsilon W)] \\
&= E_\epsilon[Y^TY - Y^TXW - Y^T\epsilon W - W^TX^TY + W^TX^TXW + W^TX^T\epsilon W \\
&\quad - W^T\epsilon^TY + W^T\epsilon^TXW + W^T\epsilon^T\epsilon W] \\
&= E_\epsilon[Y^TY - 2W^TX^TY - 2W^T\epsilon^TY + 2W^TX^T\epsilon W + W^TX^TXW \\
&\quad + W^T\epsilon^T\epsilon W] \\
&= Y^TY - 2W^TX^TY + W^TX^TXW - 2E_\epsilon[W^T\epsilon^TY] \\
&\quad + 2E_\epsilon[W^TX^T\epsilon W] + E_\epsilon[W^T\epsilon^T\epsilon W] \qquad (1)
\end{aligned}$$

And $W^T\epsilon^TY$ is nothing but $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{d+1} w_j\epsilon_{ij}y_i$

$W^TX^T\epsilon W$ is nothing but $\displaystyle\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\sum_{k=1}^{n} w_iw_jx_{ik}\epsilon_{kj}$

$W^T\epsilon^T\epsilon W$ is nothing but $\displaystyle\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\sum_{k=1}^{n} w_iw_j(\epsilon_{ik}\cdot\epsilon_{kj})$

Now computing the expectations of these 3 terms with respect to the attribute noise

$$\begin{aligned}
E_\epsilon[W^T\epsilon^TY] &= E_\epsilon[\sum_{i=1}^{n}\sum_{j=1}^{d+1} w_je_{ij}y_i] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{d+1} w_jy_iE_\epsilon[e_{ij}] \qquad \because \text{By using the linearity of expectations}
\end{aligned}$$

$$\therefore E_\epsilon[W^T\epsilon^TY] = 0 \qquad \because E_\epsilon[e_{ij}] = 0 \qquad (2)$$

Now

$$\begin{aligned}
E_\epsilon[W^TX^T\epsilon W] &= E_\epsilon[\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\sum_{k=1}^{n} w_iw_jx_{ik}\epsilon_{kj}y_i] \\
&= \sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\sum_{k=1}^{n} w_iw_jx_{ik}E_\epsilon[\epsilon_{kj}] \qquad \because \text{By using the linearity of expectations}
\end{aligned}$$

$$\therefore E_\epsilon[W^TX^T\epsilon W] = 0 \qquad (3)$$

Now,

$$\begin{aligned}E_\epsilon[W^T\epsilon^T\epsilon] &= E_\epsilon[\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\sum_{k=1}^{n} w_i w_j(\epsilon_{ik}\epsilon_{kj})] \\ &= \sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\sum_{k=1}^{n} w_i w_j E_\epsilon[\epsilon_{ik}\cdot\epsilon_{kj}] \qquad \because \text{By using the linearity of expectations}\end{aligned}$$

Since ϵ_i is I.I.D and co variance matrix of ϵ_i is $\sigma^2 I$.

$$\begin{aligned}E_\epsilon[\epsilon_{ik}\cdot\epsilon_{kj}] &= 0 \quad \text{if i} \neq \text{j} \\ &= \sigma^2 \quad \text{if i = j}\end{aligned}$$

$$\therefore E_\epsilon[W^T\epsilon^T W] = \sigma^2 \tag{4}$$

using (2), (3), (4) in equation (1) We get

$$\begin{aligned}& Y^TY - 2W^TX^TY + W^TX^TXW + \sigma^2\sum_{i=1}^{d+1} w_i^2 \\ \text{i.e } & Y^TY - 2W^TX^TY + W^TX^TXW + \sigma^2||W||_2^2 \\ \implies & Y^TY - 2W^TX^TY + W^TX^TXW + W^T\sigma^2 W\end{aligned}$$

Differentiating with respect to W and equating it to 0 we get

$$\begin{aligned}-2X^TY + 2X^TXW + 2\sigma^2 IW = 0 \\ (X^TX + \sigma^2 I)W = X^TY\end{aligned}$$

$$W = (X^TX + \sigma^2 I)^{-1}X^TY$$

Comparing this to the solution of ridge regression the λ in ridge regression here is σ^2

Question 3

We know that for a data set of size 'n' and hypothesis class 'H'

For, $n \leq \delta_H \implies G_H(n) = 2^n$
$n > \delta_H \implies G_H(n) = 2^n$
So, there are 2^n distinct ways to label the the 'n' points.

To shatter these data points we should have at least 2^n hypothesis and VC(H) = n

$$|H| \geq 2^n$$
$$n = \delta_H \leq log_2|H|$$

Question 4

Notions Used

- H(D) = Homogeneity measure of labels
- Chi value - Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.

  $Chi = \sum_{ClassesiChildrenj} \frac{(N_{ij}-E_{ij})^2}{E_{ij}}$

  1. N_{ij} = Number of points of class i in node j
  2. E_{ij} = Expected number of points of class i in node j i.e $N_i * Pj$

- Expected information needed to classify a tuple

  $E(D) = \sum_{i=1}^{m} p_i \log pi$

- Information needed (after using attribute A to split D into v partitions) to classify D

  $E(D, A) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} E(D_j)$

- Gain = E(D) - E(D,A)
- D1....Dn = partitions
- D = dataset
- n_i = no_of elements in Di
- n = no of elements in D without selected attribute missing
- probability(D) - gives the probability vector of class labels
- The probability of class labels at leaf node is the product of all the conditional probabilities of all the nodes on the path from route to leaf

Algorithm 1 Decision Tree

```
procedure PARTITIONS(D,attr)
    for  x = 1....n do
        if x[attr] is none then
            Add to Di with probability of n_i/n
        else
            D1,D2,.,,,Dn = split(D,best_attr)
procedure PROBABILITY(D)
    c_i = number of labels of class i in dataset D
    p_i = c_i / n
    C = [p_1,.....p_k]
    return C

procedure LEARNING(D)
    if H(D) = 1 then
        return probability(D)
    else if no attributes then
        return probability(D)
    else
        for  j=1...d do
            D_1,D_2...D_n = partition(D,j)
            Gain = E(D) - E(D,i)
            best_value = max(Gain,best_value)
        new_attributes = actual - best_attribute
        for item in partitions do
            add_sub_tree(learning(item,new_attributes))
        for each leaf do
            find probability(leaf)
        if node.all_leaves then
            if  significant_chi then
                return make_leaf(probability(D))

procedure CLASSIFY(point,tree)
    if point.results != None  then
        return point.results
    else
        v = values[tree.columns]
        if v == None then
            classify(point, true_branch)
            classify(point, false_branch)
            modify result[k] with corresponding probabilities        ▷ k class label
        else
            if v == tree.value then
                branch = true_branch
            else
                branch = false_branch
        return classify(point,branch)
```

Question 5

To show that 'k' is a valid kernel it is sufficient to show that there exists a feature transformation ϕ.

Let μ be the center of mass in the transformed feature space.So μ is the average of all the data points in new feature space

$$\therefore \mu = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$$

The square Euclidean distance from μ to $\phi(x)$ would be

$$\begin{aligned}
||\mu - \phi(x)||^2 &= \langle \mu - \phi(x), \mu - \phi(x)\rangle \\
&= \langle \frac{1}{n}\sum_i \phi(x_i) - \phi(x), \frac{1}{n}\sum_i \phi(x_i) - \phi(x)\rangle \\
&= \frac{1}{n^2}\sum_{i,j}\langle \phi(x_i), \phi(x_j)\rangle - \frac{2}{n}\sum_i \langle \phi(x_i), \phi(x)\rangle + \langle \phi(x), \phi(x)\rangle \\
&= \frac{1}{n^2}\sum_{i,j} k(x_i, x_j) - \frac{2}{n}\sum_i k(x, x_i) + k(x, x)
\end{aligned}$$

$$\begin{aligned}
\therefore \frac{1}{n}||\mu - \phi(x)||^2 &= \frac{1}{n}(\frac{1}{n^2}\sum_{i,j} k(x_i, x_j) - \frac{2}{n}\sum_i k(x, x_i) + k(x, x)) \\
&= \frac{1}{n^3}\sum_{i,j} k(x_i, x_j) - \frac{2}{n^2}\sum_i k(x, x_i) + \frac{1}{n}k(x, x)
\end{aligned}$$

Closure properties of kernels,

1. If k_1 & k_2 are kernel functions then $k = k_1 + k_2$ is also a kernel function.

   Let Kernel or Gram matrix K_1 correspond to Kernel function k_1 & Kernel or Gram matrix K_2 correspond to Kernel function k_2. K1 and K2 are semi positive definite.So,

$$\alpha^T K_1 \alpha \geq 0 \quad \text{and} \quad \alpha^T K_2 \alpha \geq 0$$
$$\alpha^T K_1 \alpha + \alpha^T K_2 \alpha \geq 0$$
$$\alpha^T K \alpha \geq 0$$

2. Multiplication of two kernel functions gives an kernel function

$$
\begin{aligned}
k(x, y) &= k_1(x, y) \cdot k_2(x, y) \\
&= \sum_{i=1} \phi_i(x) \cdot \phi_i(y) \cdot \sum_{j=1} \psi_j(x) \cdot \psi_j(y) \\
&= \sum_i \sum_j [\phi_i(x) \cdot \psi_j(x)] \cdot [\phi_i(y) \cdot \psi_j(y)] \\
&= \sum_i \sum_j p_{ij}(x) \cdot p_{ij}(y) \quad \text{where } p \text{ is a feature transformation} \\
&= p(x)^T \cdot p(y)
\end{aligned}
$$

By these two properties we can say that above transformation is an feature transformation.

__________ End of Document __________