CSci 5512 – Artificial Intelligence II Jiecao Chen ID:4716311
HOMEWORK #4

Question 1

a. run **python dtree4.py**
result:

```
The tree structure after learning from all example:
Pat 's children ==============
None F
Full Hun
Some T
Hun 's children ==============
T Type
F F
Type 's children ==============
Burger T
Thai Fri
French T
Italian F
Fri 's children ==============
T T
F F
Training Set Error Rate: 0.0


Now do LOOCV :
leave Example 1 out: real class - T ,by DTree - T
leave Example 2 out: real class - F ,by DTree - T
leave Example 3 out: real class - T ,by DTree - T
leave Example 4 out: real class - T ,by DTree - F
leave Example 5 out: real class - F ,by DTree - T
leave Example 6 out: real class - T ,by DTree - T
leave Example 7 out: real class - F ,by DTree - F
leave Example 8 out: real class - T ,by DTree - T
leave Example 9 out: real class - F ,by DTree - T
leave Example 10 out: real class - F ,by DTree - T
leave Example 11 out: real class - F ,by DTree - F
leave Example 12 out: real class - T ,by DTree - F
LOOCV Error Rate: 0.5
```

b. run **python dlist2.py**
result:

```
Pat Some,  ==> T , otherwise
```

```
Hun F,  ==> F , otherwise
Type Italian,  ==> F , otherwise
Fri F,  ==> F , otherwise
Alt T,  ==> T , otherwise
const F,  ==> F , otherwise

now test training dataset...
Example 1 real cls: T  by dlist: T
Example 2 real cls: F  by dlist: F
Example 3 real cls: T  by dlist: T
Example 4 real cls: T  by dlist: T
Example 5 real cls: F  by dlist: F
Example 6 real cls: T  by dlist: T
Example 7 real cls: F  by dlist: F
Example 8 real cls: T  by dlist: T
Example 9 real cls: F  by dlist: F
Example 10 real cls: F  by dlist: F
Example 11 real cls: F  by dlist: F
Example 12 real cls: T  by dlist: T
Training data error rate: 0.0

now LOOCV test...
Example 1 real cls: T  by dlist when leave 1 out: T
Example 2 real cls: F  by dlist when leave 2 out: T
Example 3 real cls: T  by dlist when leave 3 out: T
Example 4 real cls: T  by dlist when leave 4 out: T
Example 5 real cls: F  by dlist when leave 5 out: T
Example 6 real cls: T  by dlist when leave 6 out: T
Example 7 real cls: F  by dlist when leave 7 out: F
Example 8 real cls: T  by dlist when leave 8 out: T
Example 9 real cls: F  by dlist when leave 9 out: T
Example 10 real cls: F  by dlist when leave 10 out: T
Example 11 real cls: F  by dlist when leave 11 out: F
Example 12 real cls: T  by dlist when leave 12 out: F
LOOCV error rate: 0.416666666667
```

c. run **python perceptron.py**
result:

```
Example 0 True class: T , classified by NN: T
Example 1 True class: F , classified by NN: F
Example 2 True class: T , classified by NN: T
Example 3 True class: T , classified by NN: T
Example 4 True class: F , classified by NN: F
Example 5 True class: T , classified by NN: T
Example 6 True class: F , classified by NN: F
Example 7 True class: T , classified by NN: T
```

```
Example 8 True class: F , classified by NN: F
Example 9 True class: F , classified by NN: F
Example 10 True class: F , classified by NN: F
Example 11 True class: T , classified by NN: T
training dataset error rate: 0.0

Now do LOOCV ...
Example 0 True class: T , classified by NN (leave Example 0 out): F
Example 1 True class: F , classified by NN (leave Example 1 out): T
Example 2 True class: T , classified by NN (leave Example 2 out): F
Example 3 True class: T , classified by NN (leave Example 3 out): F
Example 4 True class: F , classified by NN (leave Example 4 out): F
Example 5 True class: T , classified by NN (leave Example 5 out): T
Example 6 True class: F , classified by NN (leave Example 6 out): T
Example 7 True class: T , classified by NN (leave Example 7 out): F
Example 8 True class: F , classified by NN (leave Example 8 out): F
Example 9 True class: F , classified by NN (leave Example 9 out): T
Example 10 True class: F , classified by NN (leave Example 10 out): F
Example 11 True class: T , classified by NN (leave Example 11 out): F
LOOCV error rate: 0.666666666667
```

Question 2

a. To show this, we just need to show every hypothesis in (k-DT(n)) could be represented as a hypothesis in (k-DL(n)).

Proof:

suppose $h \in$ (k-DT(n)), for each leaf in h, say l_i there is a path from root to it, now we construct conjunction by the test-value pairs in the path, say c_i, suppose we have N leaves in h, then the following decision-list is equivalent to h

```
C_1? -No-> C_2? -No-> C_3 -No-> ... -No-> C_N? -No-> !L_N
|yes       |yes       |yes                |yes
L_1        L_2        L_3                 L_4
```

b. k is a constant, for the k-DT(n), consider the Full Binary Tree with depth k, there are at most $m = 2^{k+1}-1$ nodes, and each node has at most n choices, thus $|\text{k-DT}(n)| = O(n^m)$ For TLF(n), suppose $n \gg m$,

Claim:

For each x in $\{0,1\}^n$, there is a function in TLF(n) that can classify $\{x\}$ and $\{0,1\}^n \setminus \{x\}$ (I will show this later)

Since $|\{0,1\}^n| = 2^n$, we know that $|\text{TLF}(n)| = o(2^n)$, thus $|\text{k-DT}(n)| = O(n^m) = O(2^n)$ is smaller than $|\text{TLF}(n)|$

Now come to prove the claim:

Give x in $\{0,1\}^n$, let $w = x$, let $b = w^T x - \epsilon$ where $1 \gg \epsilon > 0$, now we can see that

$w^T x - b = \epsilon > 0$, if $y \neq x$, then $w^T y - b \leq -1 + \epsilon < 0$, so we know that the claim is true since we can construct such a w

Question 3

suppose b_0 is the $(n+1)_{th}$ highest bid we can define the utility function as

$$u_i = \begin{cases} v_i - b_0 & b_i > b_0 \\ 0 & b_i \leq b_0 \end{cases} \tag{1}$$

consider b_0 as a r.v. Now we consider the company i.
case 1:
$E[u_i|\{b_0 \leq v_i\}] = P(\{b_i > b_0\}|\{b_0 \leq v_i\})(v_i - b_0)$
note that $(v_i - b_0) > 0$ and when $b_i \leq v_i$, $P(\{b_i > b_0\}|\{b_0 \leq v_i\})$ would increase when b_i increases.
when $b_i > v_i$, $P(\{b_i > b_0\}|\{b_0 < v_i\}) = P(\{v_i > b_0\}|\{b_0 < v_i\})$ so, $b_i \geq v_i$ the best choice in this case.
case 2:
$E[u_i|\{b_0 > v_i\}] = P(\{b_i > b_0\}|\{b_0 < v_i\})(v_i - b_0)$
note that if $b_i > v_i$, $E[u_i|\{b_0 > v_i\}] < 0$, otherwise, $E[u_i|\{b_0 > v_i\}] = 0$. so in this case $b_i \leq v_i$ would be the best choices.
combine two cases, $E[u_i] = P(\{b_0 > v_i\})E[u_i|\{b_0 > v_i\}] + P(\{b_0 \leq v_i\})E[u_i|\{b_0 \leq v_i\}]$ we know that $b_i = v_i$ is the optimal bid.

b. (i) is true,(ii) is absolutely wrong.
Suppose b_0 is the $(n+1)_{th}$ highest bid when company C only put one bid. assume $b_0 \ll v$, in this case, it would make profit of $v - b_0$
Now if it put two bids as described, it is possible that $v - \epsilon$ may become the $(n+1)_{th}$ highest bid, then in this case, C only makes profit of ϵ, which is much smaller than $v - b_0$.

Question 4

a.

$$f_1(W) = \sum_{i=1}^{n} (y_i - W^T x_i)^2$$

let $p \in (0,1), q = (1-p)$

$$\begin{aligned}
f(pW_1 + qW_2) &= \sum \left(y_i - (pW_1 + qW_2)^T x_i\right)^2 \\
&= \sum (p\Delta_{1i} + q\Delta_{2i})^2 \\
&= \sum (p^2\Delta_{1i}^2 + q^2\Delta_{2i}^2 + 2pq\Delta_{1i}\Delta_{2i}) \\
&\le \sum [p^2\Delta_{1i}^2 + q^2\Delta_{2i}^2 + pq(\Delta_{1i}^2 + \Delta_{2i}^2)] \\
&= \sum (p\Delta_{1i}^2 + q\Delta_{2i}^2) \\
&= pf_1(W_1) + qf_1(W_2)
\end{aligned}$$

here $\Delta_{ki} = y_i - W_k^T x_i$, so $f_1(W)$ is a convex function by the definition.

b. Note that if $k(x), g(x)$ are both convex functions, then so is $(f+g)(x)$.So we only need to prove $g(w) = \|W\|_1$ is a convex function.

$$\begin{aligned}
g(pW_1 + qW_2) &= \|pW_1 + qW_2\|_1 \\
&\le \|pW_1\|_1 + \|qW_2\|_1 \\
&= p\|W_1\|_1 + q\|W_2\|_1 \\
&= pg(W_1) + qg(W_2)
\end{aligned}$$

So, $g(W)$ is a convex function. So is $f_2(W)$

Question E.1

a. Suppose that experts only vote **Yes** or **No** Assume that we do not make decision when number of Yes is equal to number of No.
Let N be the number of correct votes, then

$$\begin{aligned}
\text{error} &= P(N \le \lfloor \frac{M-1}{2} \rfloor) \\
&= \sum_{i=0}^{\lfloor \frac{M-1}{2} \rfloor} \binom{M}{i} (1-\epsilon)^i \epsilon^{M-i}
\end{aligned}$$

b.

$\epsilon \setminus M$	5	10	20
0.1	0.00856	0.000146903	$7.088606e^{-7}$
0.2	0.05792	0.00636938	0.000563414
0.4	0.31744	0.166239	0.127521

c. Yes, it is possible, here is an example:
Suppose $M = 3$ and $\epsilon = 1/3$. Suppose we have 6 examples.
e1: M1 and M2 are in error,
e2: M1 and M3 are in error,
e3: M2 and M3 are in error,
e4, e5, e6: None of the experts make error.
so e1,e2,e3 will be in error, and e4,e5,e6 will be predicted correctly.
the overall error rate $1/2 > 1/3 = \epsilon$