Homework #6

RELEASE DATE: 12/23/2021

DUE DATE: 01/06/2022, BEFORE 13:00 on Gradescope

QUESTIONS ARE WELCOMED ON THE NTU COOL FORUM.

You will use Gradescope to upload your choices and your scanned/printed solutions. For problems marked with (*), please follow the guidelines on the course website and upload your source code to Gradescope as well. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 16 problems and a total of 400 points. For each problem, there is one correct choice. If you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get 0 points. For four of the secretly-selected problems, the TAs will grade your detailed solution in terms of the written explanations and/or code based on how logical/clear your solution is. Each of the four problems graded by the TAs counts as additional 20 points (in addition to the correct/incorrect choices you made). In general, each homework (except homework 0) is of a total of 400 points.

Aggregation

1. Consider an aggregation classifier G constructed by uniform blending on 11 classifiers $\{g_t\}_{t=1}^{11}$. That is,

$$G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{11} g_t(\mathbf{x})\right).$$

d

Assume that each g_t is of test 0/1 error $E_{\text{out}}(g_t) = e_t$. Which of the following is the tightest upper bound of $E_{\text{out}}(G)$? Choose the correct answer; explain your answer.

[a] $\frac{1}{3}\sum_{t=1}^{11} e_t$

[b] $\frac{1}{4}\sum_{t=1}^{11} e_t$

[c] $\frac{1}{6}\sum_{t=1}^{11} e_t$

[d] $\frac{1}{11}\sum_{t=1}^{11} e_t$

[e] $\frac{1}{12}\sum_{t=1}^{11} e_t$

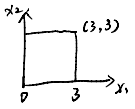
Bound $(g_t) = e_t$
for 0/1 test
K binary trees will go wrong
$\sum_{k=1}^{K} e_k \; N/K$
$E_{out}(G) \le \frac{1}{K}\sum_{k=1}^{K} e_k$

2. Suppose that each $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ is drawn uniformly from the region

$$\{0 \le x_1 \le 3, 0 \le x_2 \le 3\}$$

and the target function is $f(\mathbf{x}) = \text{sign}(x_2 - x_1)$. Consider blending the following three hypotheses linearly to approximate the target function.

$$\begin{aligned} g_1(\mathbf{x}) &= \text{sign}(x_1 - 2) \\ g_2(\mathbf{x}) &= \text{sign}(x_2 - 1) \\ g_3(\mathbf{x}) &= \text{sign}(x_2 - 2) \end{aligned}$$



That is,

d

$$G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{3} \alpha_t \cdot g_t(\mathbf{x})\right)$$

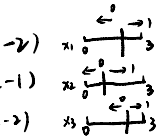
with $\alpha_t \in \mathbb{R}$. What is the smallest possible $E_{\text{out}}(G)$? Choose the correct answer; explain your answer.

(*Hint: The "boundary" of G must be a "combination" of the boundaries of g_t*)

[a] $\frac{6}{18}$
[b] $\frac{5}{18}$
[c] $\frac{4}{18}$
[d] $\frac{3}{18}$
[e] none of the other choices

E_{out} for G is calculated by $\frac{2}{N+1}\sum_{k=1}^{k} e_k$ as upper bound

$\begin{cases} g_1(x) = \text{sign}(x_1 - 2) \\ g_2(x) = \text{sign}(x_2 - 1) \\ g_3(x) = \text{sign}(x_2 - 2) \end{cases}$



boundaries of g_t

$G_{t+1} = \frac{1}{3} \times \frac{1}{3} \times \frac{3}{2} = \frac{3}{18}$

3. When talking about non-uniform voting in aggregation, we mentioned that $\boldsymbol{\alpha}$ can be viewed as a weight vector learned from any linear algorithm coupled with the following transform:

$$\boldsymbol{\phi}(\mathbf{x}) = \Big(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_T(\mathbf{x})\Big).$$

When studying kernel methods, we mentioned that the kernel is simply a computational short-cut for the inner product $(\boldsymbol{\phi}(\mathbf{x}))^T(\boldsymbol{\phi}(\mathbf{x}'))$. In this problem, we mix the two topics together using the decision stumps as our $g_t(\mathbf{x})$.

Assume that the input vectors contain only even integers between (including) $2L$ and $2R$, where $L < R$. Consider the decision stumps $g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}\Big(x_i - \theta\Big)$, where

$$\begin{aligned} i &\quad \in \{1, 2, \cdots, d\}, \\ d &\quad \text{is the finite dimensionality of the input space,} \\ s &\quad \in \{-1, +1\}, \\ \theta &\quad \text{is an odd integer between } (2L, 2R). \end{aligned}$$

sign $(x_i - \theta)$: $(R-L)$ classes

Define $\boldsymbol{\phi}_{ds}(\mathbf{x}) = \Big(g_{+1,1,2L+1}(\mathbf{x}), g_{+1,1,2L+3}(\mathbf{x}), \ldots, g_{+1,1,2R-1}(\mathbf{x}), \ldots, g_{-1,d,2R-1}(\mathbf{x})\Big)$. What is $K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}'))$? Choose the correct answer; explain your answer.

a

[a] $2d(R-L) - 2\|\mathbf{x} - \mathbf{x}'\|_1$
[b] $2d(R-L)^2 - 2\|\mathbf{x} - \mathbf{x}'\|_1^2$
[c] $2d(R-L) - 2\|\mathbf{x} - \mathbf{x}'\|_2$
[d] $2d(R-L)^2 - 2\|\mathbf{x} - \mathbf{x}'\|_2^2$
[e] none of the other choices

$g_t(x)\, g_t(x') = (s_t \cdot \text{sign}(x_i - \theta_t))(s_t \cdot \text{sign}(x_i' - \theta_t))$
$= \text{sign}(x_{t_i} - \theta_t)\, \text{sign}(x_{t_i}' - \theta_t)$

$K_{ds}(x, x') = (\phi_{ds}(x))^T \phi_{ds}(x')$
$= \sum_{t=1}^{|G|} g_t(x)\, g_t(x')$
$= \sum_{t=1}^{|G|} \text{sign}(x_{t_i} - \theta_t)\, \text{sign}(x_{t_i}' - \theta_t)$

$\text{sign}(x_{t_i} - \theta_t)\, \text{sign}(x_{t_i}' - \theta_t) = \begin{cases} -1, & \theta_t \in [\min(x_{t_i}, x_{t_i}'), \max(x_{t_i}, x_{t_i}')) \\ +1, & \text{otherwise} \end{cases}$

when $s = -1$.
$\sum_{i=1}^{d} |x_j - x_j'| = 2\|x - x'\|_1$
$\sum_{t=1}^{|G|} |\text{sign}(x_{t_i} - \theta_t)\, \text{sign}(x_{t_i}' - \theta_t)| = |G|$
$K_{ds} = \sum_{t=1}^{|G|} \text{sign}(x_{t_i} - \theta_t)\, \text{sign}(x_{t_i}' - \theta_t)$
$= |G| - 2\|x - x'\|_1 - 2\|x - x'\|_1$
$= 2d(R-L) - 4\|x - x'\|_1 + 2$

therefore, from $2L+1, \ldots, 2R-1$
special case : $2d(R-L) - 2\|x - x'\|_1$

Adaptive Boosting

4. Consider applying the AdaBoost algorithm on a binary classification data set where 99% of the examples are positive. Because there are so many positive examples, the base algorithm within AdaBoost returns a constant classifier $g_1(\mathbf{x}) = +1$ in the first iteration. Let $u_n^{(2)}$ be the individual example weight of each example in the second iteration. What is

$$\frac{\sum_{n:\ y_n>0} u_n^{(2)}}{\sum_{n:\ y_n<0} u_n^{(2)}}?$$

b

Choose the correct answer; explain your answer.

[a] 99
[b] 1/99
[c] 1
[d] 100
[e] 1/100

on negative examples $\frac{u_+^{(2)}}{u_-^{(2)}} = \frac{1}{99}$

100 ⇒ 99 "+", 1 "−"

5. For the AdaBoost algorithm introduced in Lecture 12, let $G_t(\mathbf{x}) = \text{sign}\left(\sum_{\tau=1}^{t} g_\tau(\mathbf{x})\right)$. How many of the following are guaranteed to be non-increasing from the t-th iteration to the $(t+1)$-th iteration? Choose the correct answer; explain each non-increasing case within your answer.

- $E_{\text{in}}(G_t)$ to $E_{\text{in}}(G_{t+1})$ ✗ can be unstable
- $E_{\text{out}}(G_t)$ to $E_{\text{out}}(G_{t+1})$ ✗ can be unstable.
- $\sum_{n=1}^{N} u_n^{(t)}$ to $\sum_{n=1}^{N} u_n^{(t+1)}$ ① $U^{t+1} = U^t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)}$ ↓ when $0 < \epsilon_t < \frac{1}{2}$
- $u_n^{(t)}$ to $u_n^{(t+1)}$ when g_t is correct on $(\mathbf{x}_n, y_n)$ ✓ ② as t increasing, U^t goes down as deformation
- $u_n^{(t)}$ to $u_n^{(t+1)}$ when g_t is incorrect on $(\mathbf{x}_n, y_n)$ ✗

b

up & down
up & down non-increasing guanranteed
iteration increasing

[a] 1
[b] 2
[c] 3
[d] 4
[e] 5

6. For the AdaBoost algorithm introduced in Lecture 12, let $U_t = \sum_{n=1}^{N} u_n^{(t)}$. Assume that $0 < \epsilon_t < \frac{1}{2}$ for each hypothesis g_t. What is $\frac{U_{t+1}}{U_t}$? Choose the correct answer; explain your answer.

b

[a] $\sqrt{\epsilon_t(1-\epsilon_t)}$
[b] $2\sqrt{\epsilon_t(1-\epsilon_t)}$
[c] $\sqrt{\frac{\epsilon_t}{(1-\epsilon_t)}}$
[d] $\ln\sqrt{\frac{(1-\epsilon_t)}{\epsilon_t}}$
[e] $\ln\sqrt{\frac{\epsilon_t}{(1-\epsilon_t)}}$

$$\begin{cases} u_n^{t+1} = u_n^t \, e^{-y_n \alpha_t g_t(x_n)} \\ \epsilon_t = \dfrac{\sum_{y_n \neq g_t(x_n)} u_n^t}{\sum_{n=1}^{N} u_n^t} \\ e^{\alpha_t} = \sqrt{\dfrac{1-\epsilon_t}{\epsilon_t}} \end{cases}$$

$$\begin{aligned} U_{t+1} &= \sum_{n=1}^{N} u_n^{t+1} \\ &= \sum_{n=1}^{N} u_n^t e^{-y_n \alpha_t g_t(x_n)} \\ &= \sum_{y_n = g_t(x_n)} u_n^t e^{-\alpha_t} + \sum_{y_n \neq g_t(x_n)} u_n^t e^{\alpha_t} \\ &= \left(\sum_{n=1}^{N} u_n^t\right)\left(e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t\right) \\ &= U_t\left(\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}(1-\epsilon_t) + \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\epsilon_t\right) \\ &= 2U_t\sqrt{\epsilon_t(1-\epsilon_t)} \end{aligned}$$

7. Following the previous two problems, assume that $\epsilon_t \leq \epsilon < \frac{1}{2}$, which of the following is the tightest upper bound on the number of iterations T required to ensure $E_{\text{in}}(G_T) = 0$? Choose the correct answer; explain your answer.

(*Hint: use the fact that*

$$\sqrt{\epsilon(1-\epsilon)} \leq \frac{1}{2}\exp\left(-2(\frac{1}{2}-\epsilon)^2\right)$$

for all $0 < \epsilon < \frac{1}{2}$).

b

$e^{\alpha_t} = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$

$U_t = \sum_{n=1}^{N} u_n^{(t)}$

$\frac{U_{t+1}}{U_t} = 2\sqrt{\epsilon_t(1-\epsilon_t)}$

Find T s.t. $E_{in}(G_T) = 0$

due to $0 < \epsilon < \frac{1}{2}$

$N < \exp(-2(\frac{1}{2}-\epsilon)^2) \cdot e^T$

$T > \frac{\ln N}{2(\frac{1}{2}-\epsilon)^2}$

- [a] $\frac{\ln N}{2(\frac{1}{2}-\epsilon)}$
- [b] $\frac{\ln N}{2(\frac{1}{2}-\epsilon)^2}$
- [c] $\frac{\ln N}{4(\frac{1}{2}-\epsilon)}$
- [d] $\frac{\ln N}{4(\frac{1}{2}-\epsilon)^2}$
- [e] $\frac{\ln N}{4(\frac{1}{2}-\epsilon)^4}$

Random Forest = Bagging + Decision Tree

8. Suppose we have a data set of size $N = 1126$, and we use bootstrapping to sample N' examples. What is the minimum N' such that the probability of getting at least one duplicated example (with # copies ≥ 2) is larger than 50%? Choose the correct answer; explain your answer.

d

$N = 1126$

$p(\text{1 duplicated examples}) > 50\%$

$N' = pN$

$E_{out}(G) \leq \frac{2}{K+1} \sum_{k=1}^{K} \epsilon_k$

$\frac{2 \times 40}{1126+1} \leq \frac{1}{2} \times \frac{1}{100} = 50\% \cdot 1.8\%$

$\Rightarrow N' = 40$

- [a] 25
- [b] 30
- [c] 35
- [d] 40
- [e] none of the other choices

9. If bootstrapping is used to sample exactly $2N$ examples out of N, what is the probability that an example is *not* sampled when N is very large? Choose the closest answer; explain your answer.

d

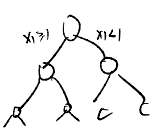
$e^{-p} \cdot N$, $p = 2$ we get $\frac{1}{e^2} \approx 13.5\%$

$(1-\frac{1}{N})^{N'} = (1-\frac{1}{N})^{pN} = [(1-\frac{1}{N})^N]^p \approx e^{-p}$

- [a] 77.9%
- [b] 60.7%
- [c] 36.8%
- [d] 13.5%
- [e] 1.8%

10. Suppose we have a set of decision trees. Each tree comes with 2 node, each equipped with a fixed branching function. The root node is of two branches, evaluating whether $x_1 \geq 0$. If $x_1 < 0$, the node connects to a leaf with some constant output. Otherwise the node connects to another node of two branches, evaluating whether $x_2 \geq 0$. Each of the branches connects to a constant leaf. Consider three-dimensional input vectors. That is, $\mathbf{x} = (x_1, x_2, x_3)$. Which of the following data set can be shattered by the set of decision trees? Choose the correct answer; explain your answer.

b



$\chi = (x_1, x_2, x_3)$

$(1,1,-1)$
$(-1,1,-1)$ $\Rightarrow$ $(1,1,-1)$ $\Rightarrow$ $(1,1,-1)$
$(1,-1,-1)$

- [a] $(1,1,-1),(-1,1,-1),(-1,-1,-1)$
- [b] $(1,1,-1),(-1,1,-1),(1,-1,-1)$
- [c] $(1,1,-1),(-1,1,-1),(1,-1,-1),(-1,-1,-1)$
- [d] $(1,1,-1),(-1,1,-1),(1,-1,-1),(-1,-1,1)$
- [e] none of the other choices

Experiments with Adaptive Boosting

For Problems 11-16, implement the AdaBoost-Stump algorithm as introduced in Classes 12 and 13. Run the algorithm on the following set for training:

`https://www.csie.ntu.edu.tw/~htlin/course/ml21fall/hw6/hw6_train.dat`

and the following set for testing:

`https://www.csie.ntu.edu.tw/~htlin/course/ml21fall/hw6/hw6_test.dat`

Use a total of $T = 500$ iterations (please do not stop earlier than 500), and calculate E_{in} and E_{out} with the 0/1 error.

For the decision stump algorithm, please implement the following steps. Any ties can be arbitrarily broken.

(1) For any feature i, sort all the $x_{n,i}$ values to $x_{[n],i}$ such that $x_{[n],i} \leq x_{[n+1],i}$.

(2) Consider thresholds within $-\infty$ and all the midpoints $\frac{x_{[n],i}+x_{[n+1],i}}{2}$. Test those thresholds with $s \in \{-1, +1\}$ to determine the best (s, θ) combination that minimizes E_{in}^u using feature i.

(3) Pick the best (s, i, θ) combination by enumerating over all possible i.

For those interested in algorithms (who isn't? :-)), step 2 can be carried out in $O(N)$ time only!!

11. (*) What is the value of $E_{\text{in}}(g_1)$? Choose the closest answer; provide your code.

c based on code

[a] 0.29
[b] 0.33
[c] 0.37
[d] 0.41
[e] 0.45

12. (*) What is the value of $\max_{1 \leq 500 \leq t} E_{\text{in}}(g_t)$? Choose the closest answer; provide your code.

e

[a] 0.40
[b] 0.45
[c] 0.50
[d] 0.55
[e] 0.60

13. (*) What is the smallest t within the choices below such that $\min_{1 \leq \tau \leq t} E_{\text{in}}(G_\tau) \leq 0.05$? Choose the correct answer; provide your code.

d

[a] 60
[b] 160
[c] 260
[d] 360
[e] 460

14. (*) What is the value of $E_{\text{out}}(g_1)$? Choose the closest answer; provide your code.

b

[a] 0.40
[b] 0.45
[c] 0.50
[d] 0.55
[e] 0.60

15. (*) Define $G_{\text{uniform}}(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} g_t(\mathbf{x})\right)$. What is the value of $E_{\text{out}}(G_{\text{uniform}})$? Choose the closest answer; provide your code.

[a] 0.23
[b] 0.28
[c] 0.33
[d] 0.38
[e] 0.43

a

16. (*) What is the value of $E_{\text{out}}(G_{500})$? Choose the closest answer; provide your code.

[a] 0.14
[b] 0.18
[c] 0.22
[d] 0.26
[e] 0.30

b

HW6程式採用python語言，以Jupiter notebook形式呈現

```
import math
import numpy as np
import matplotlib pyplot as plt
def ReadData dataFile
    with open dataFile 'r' as f
        lines = f readlines
        data_list =
        for line in lines
            line = line strip  split
            data_list append  float l  for l in line
        dataArray = np array data_list
        return dataArray

def sign n

    if n>=0
        return 1
    else
        return -1

def GetSortedArray dataArray i
    # 根据dataArray第i列的值对dataArray进行从小到大的排序
    data_list=dataArray
    sorted_data_list=sorted data_list key=lambda x x i reverse=False
    sortedDataArray=np array sorted_data_list
    return sortedDataArray
```

```
def GetUZeroOneError pred dataY u
    return np sum u*np                pred dataY   /np sum u


def GetZeroOneError pred dataY
    return
np sum np                 pred dataY   /dataY          0

def decision_stump dataArray u

    num_data=dataArray          0
    num_dim=dataArray          1 -1
    min_e=np inf
    min_s = np inf
    min_d=np inf
    min_theta = np inf
    min_pred = np zeros   num_data
    for d in range num_dim
        sortedDataArray=GetSortedArray dataArray d  # 确
保有效theta
        d_min_e=np inf
        d_min_s = np inf
        d_min_theta = np inf
        d_min_pred = np zeros   num_data
        for s in  -1.0 1.0
            for i in range num_data
                if i==0
                    theta=-np inf
                    pred=s*np ones   num_data
                else
                    if sortedDataArray i-
1 d ==sortedDataArray i d
```

```
                    continue
                theta= sortedDataArray i-
1 d +sortedDataArray i d   /2
                pred=np zeros   num_data
                for n in range num_data
                    pred n =s*sign dataArray n d -theta

d_now_e=GetUZeroOneError pred dataArray    -1   u
            if d_now_e<d_min_e
                d_min_e=d_now_e
                d_min_s=s
                d_min_theta=theta
                d_min_pred=pred
        if d_min_e<min_e
            min_e=d_min_e
            min_s=d_min_s
            min_d=d
            min_theta=d_min_theta
            min_pred=d_min_pred
    return min_s min_d min_theta min_pred min_e
def Pred paraList dataX
    # paraList=[s,d,theta]
    num_data=dataX           0
    pred=np zeros   num_data
    for i in range num_data
        pred i =paraList 0 *sign dataX i paraList 1   -
paraList 2
    return pred


def
plot_line_chart X=np arange 0 500 1              Y=np aran
ge 0 500 1              nameX="t" nameY="Ein(gt)" saveNa
```

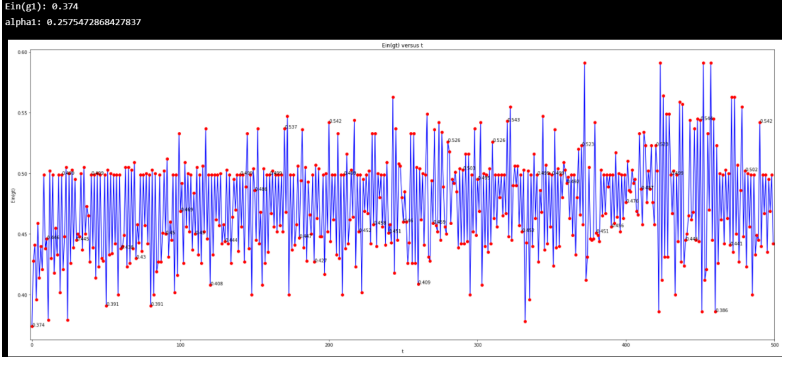
```
me="12.png"

    plt figure figsize= 30 12
    plt plot X Y 'b'
    plt plot X Y 'ro'
    plt xlim   X 0 -1 X -1 +1
    for  x y  in zip X Y
        if x%10==0
            plt text x+0.1 y str round y 4
    plt xlabel nameX
    plt ylabel nameY
    plt title nameY+" versus "+nameX
    plt savefig saveName
    return
dataArray=ReadData "hw6_train.dat.txt"
dataY=dataArray   -1
dataX=dataArray    -1
num_data=dataArray          0
u=np full shape= num_data   fill_value=1/num_data
ein_g_list=
alpha_list=
g_list=
ein_G_list=
u_sum_list=
epi_list=
min_pred_list=
# adaboost
for t in range 500
    u_sum_list append np sum u

min_s min_d min_theta min_pred epi=decision_stump d
ataArray u
    g_list append   min_s min_d min_theta
```

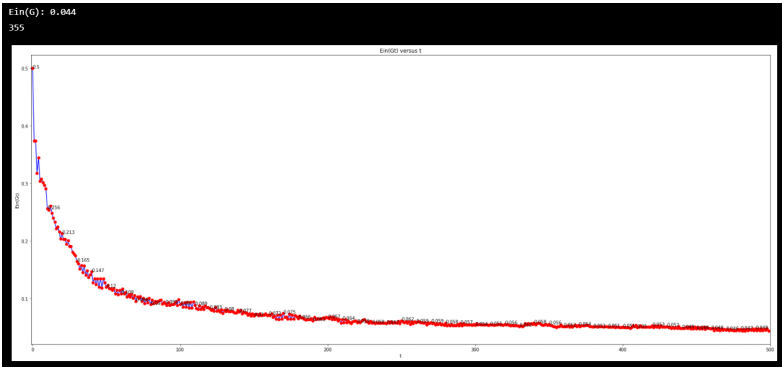
```
    min_pred_list append min_pred
    ein_g=GetZeroOneError min_pred dataY
    ein_g_list append ein_g
    epi_list append epi
    para=math sqrt  1-epi /epi
    alpha_list append math log para
    for i in range num_data
        if min_pred i ==dataY i
            u i /=para
        else
            u i *=para
    predG=np zeros  num_data
    for ta in range t
        predG+=alpha_list ta *min_pred_list ta
    for n in range num_data
        predG n =sign predG n
     ein_G_list append GetZeroOneError predG dataY

#Problem 11 and Problem 12
plot_line_chart Y=ein_g_list
print "Ein(g1):" ein_g_list 0
print "alpha1:" alpha_list 0
```




```
# Problem 13
```

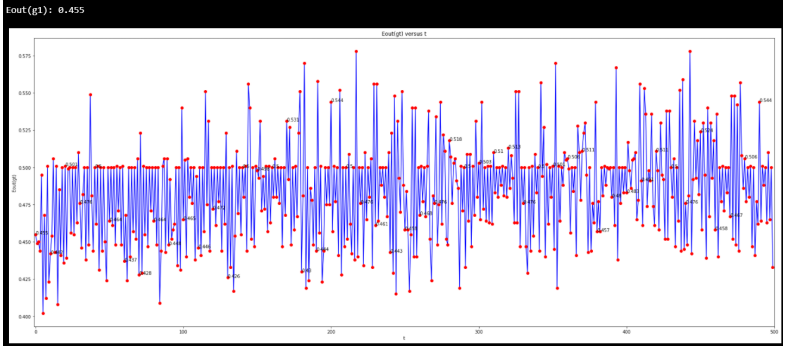
```
plot_line_chart Y=ein_G_list nameY="Ein(Gt)" saveNa
me="14.png"
print "Ein(G):" ein_G_list -1
for i in range 500
    if ein_G_list i ==0.05
        print i
        break
```



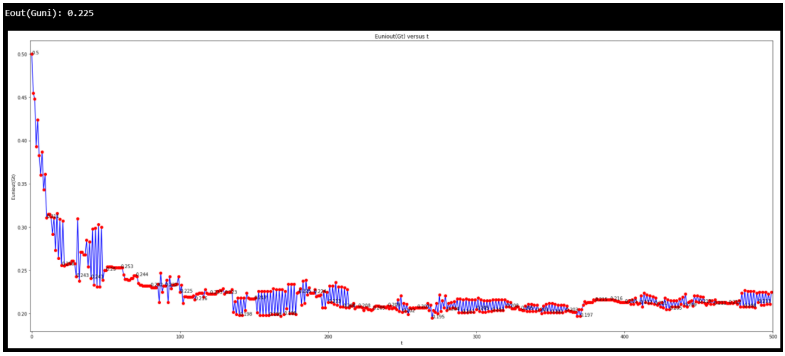

```
#計算Eout
testArray=ReadData "hw6_test.dat.txt"
num_test=testArray          0
testX=testArray    -1
testY=testArray   -1
pred_g_list=
eout_g_list=
eout_G_list=
eout_Guni_list=
for t in range 500
    pred_g=Pred g_list t  testX
    pred_g_list append pred_g
    eout_g_list append GetZeroOneError pred_g testY
    pred_G=np zeros   num_test
    pred_Guni=np zeros   num_test
```

```
    for ta in range t
        pred_G+=alpha_list ta *pred_g_list ta
    for ta in range t
        pred_Guni+=pred_g_list ta
sign_ufunc=np                    sign 1 1
pred_G=sign_ufunc pred_G
pred_Guni=sign_ufunc pred_Guni
eout_G_list append GetZeroOneError pred_G testY
eout_Guni_list append GetZeroOneError pred_Guni test
Y

#Problem 14
plot_line_chart Y=eout_g_list  nameY="Eout(gt)"
saveName="17.png"
print "Eout(g1):" eout_g_list 0
```




```
#Problem 15
plot_line_chart Y=eout_Guni_list
nameY="Euniout(Gt)"  saveName="19.png"
print "Eout(Guni):" eout_Guni_list -1
```




```
#Problem 16
plot_line_chart Y=eout_G_list  nameY="Eout(Gt)"
saveName="18.png"
print "Eout(G):" eout_G_list -1
```