Homework #3

Answer Sheet

DEADLINE: 05/23/2017, 14:00
INSTRUCTOR: Hsuan-Tien Lin

王冠鈞 b03902027

1. To let the weighted-E_{in} optimization problem equivalent to a usual E_{in} optimization problem, that is,

$$(P_1)\ \min_{\mathbf{w}} E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(\tilde{y}_n - \mathbf{w}^T\tilde{\mathbf{x}}_n)^2$$

, we can observe the original problem, and we can find that it can be written as:

$$(P_2)\ \min_{\mathbf{w}} E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}((\sqrt{u_n}y_n) - \mathbf{w}^T(\sqrt{u_n}\mathbf{x}_n))^2$$

Apparently, the condition that $P_1 \equiv P_2$ is:

$$(\tilde{\mathbf{x}}_n, \tilde{y}_n) = (\sqrt{u_n}\mathbf{x}_n, \sqrt{u_n}y_n),\ \text{for}\ n = 1, 2, \cdots, N$$

2. In the first iteration, $u_+^{(1)} = u_-^{(1)}$, and $\epsilon_1 = \frac{\sum_{n=1}^{N} u_n^{(1)}[\![y_n \neq g_1(\mathbf{x}_n)]\!]}{\sum_{n=1}^{N} u_n^{(1)}} = \frac{0.01N\times u_-^{(1)}}{0.99N\times u_+^{(1)}+0.01N\times u_-^{(1)}} = 0.01$. According to the algorithm, after updating the weights, we get:

$$u_+^{(2)} = u_+^{(1)} \div \sqrt{\frac{1-\epsilon_1}{\epsilon_1}} = u_+^{(1)}\sqrt{\frac{0.01}{0.99}}$$
$$u_-^{(2)} = u_-^{(1)} \times \sqrt{\frac{1-\epsilon_1}{\epsilon_1}} = u_-^{(1)}\sqrt{\frac{0.99}{0.01}}$$

And finally, we get

$$\frac{u_+^{(2)}}{u_-^{(2)}} = \frac{u_+^{(1)}\sqrt{\frac{0.01}{0.99}}}{u_-^{(1)}\sqrt{\frac{0.99}{0.01}}} = \frac{u_+^{(1)}\times 0.01}{u_-^{(1)}\times 0.99} = \frac{1}{99}$$

3. First, regardless of the number of dimensions, there are universally 2 decision stumps that return 1/-1 , respectively, for every input. Then, for every dimension, all data points are on the integer point in interval $[L, R]$, and there are $R - L$ spaces in the interval, and every space can accommodate 2 "different" decision stumps (positive/negative rays) that will output 1 for some inputs and -1 for others. Thus, in general, there are $2 + 2d \times (R - L)$ different decision stumps in total, and in this question $(d = 2, L = 1, R = 6)$, there are $2 + 2 \times 2 \times (6 - 1) = 22$ different stumps.

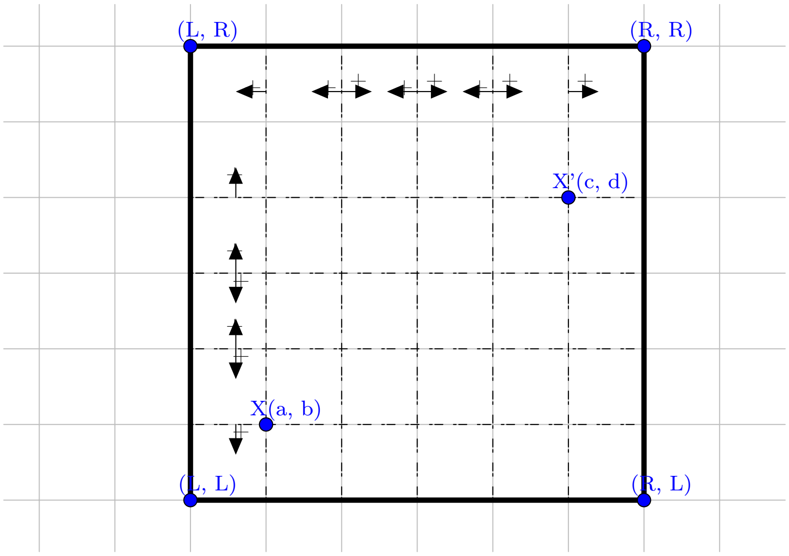
4. First, since all $g_i(\mathbf{x})$ returns either 1 or -1, and the kernel is equal to:

$$K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}')) = \sum_{t=1}^{|\mathcal{G}|} g_t(\mathbf{x})g_t(\mathbf{x}')$$

, which is a sum of 1-s and -1-s. The sign depends on whether $g_t(\mathbf{x})$ and $g_t(\mathbf{x}')$ are both positive/negative or not. We first notice that when $g(\mathbf{x}) = 1$ or $g(\mathbf{x}) = -1$, the products will definitely be positive, while in other cases it depends. We can consider the kernel function is equivalent to the following:

$$\begin{aligned} K_{ds}(\mathbf{x}, \mathbf{x}') &= \text{ \# of all different decision stumps } - 2 \times \text{ \# of decision stumps that will separate } \mathbf{x} \text{ and } \mathbf{x}' \\ &= |\mathcal{G}| - 2 \times \text{ \# of decision stumps that will separate } \mathbf{x} \text{ and } \mathbf{x}' \end{aligned}$$

The main concept is that we first assume that all $g_t(\mathbf{x})g_t(\mathbf{x}')$ are positive, giving us $|\mathcal{G}|$. However, the fact is that there may be some t such that $g_t(\mathbf{x})g_t(\mathbf{x}') = -1$, but since we previously added them as 1, we have to subtract the number of negative terms by 2 times.
Now we need to find the # of decision stumps that will separate $\mathbf{x}$ and $\mathbf{x}'$. For convenience to explain, consider a 2-dimensional space with $R - L = 6$:



In this figure, it has explicitly illustrated that the decision stumps that will separate the two points (the dot-dash lines segments) are within the intervals $[a, c]$ in the 1st dimension and $[b, d]$ in the 2nd dimension.[1] Thus the number of such decision stumps in this example is $2 \times ((c - a) + (d - b)) = 2d_1(\mathbf{x}, \mathbf{x}')$, where $d_1(\mathbf{x}, \mathbf{x}')$ is the L_1 distance[2] of the two vectors. This can be expanded to any finite dimensions without modification. Finally, the overall formula of the kernel can be written as:

$$\begin{aligned} K_{ds}(\mathbf{x}, \mathbf{x}') &= |\mathcal{G}| - 2 \times 2d_1(\mathbf{x}, \mathbf{x}') \\ &= 2 + 2d \times (R - L) - 4d_1(\mathbf{x}, \mathbf{x}') \end{aligned}$$

In the case of the previous problem, $K_{ds}(\mathbf{x}, \mathbf{x}') = 22 - 4d_1(\mathbf{x}, \mathbf{x}')$.
As for the "bonus" part for non-integers, without any other conditions I don't think that there's a solution since the number of decision stumps will become infinite, and it's meaningless to compare the difference between the infinite number of decision stumps and the infinite number of the decision stumps that will separate the two input vectors. (Unless we still use the "finite" decision stumps with the "integer" constraint, and the answer will be $2d \times (R - L) - 4d_1(\lfloor \mathbf{x} \rfloor, \lfloor \mathbf{x}' \rfloor)$, where $\lfloor \mathbf{x} \rfloor$ is the floor operation of each term in $\mathbf{x}$.)

[1] The arrows denotes the positive direction of a decision stump.

[2] Also called the *taxicab distance*, l_1 *norm* or the *Manhattan distance*. See the following reference: `https://en.wikipedia.org/wiki/Taxicab_geometry`.

5. We can simply get $1-\mu_+^2-\mu_-^2 = 1-\mu_+^2-(1-\mu_+)^2 = 2(\mu_+-\mu_+^2)$. We can find its minimum by differentiating:

$$\begin{aligned} & \frac{d}{d\mu_+}2(\mu_+-\mu_+^2) = 2-4\mu_+ = 0 \Rightarrow \mu_+ = \frac{1}{2} \\ \Rightarrow \quad & \min(1-\mu_+^2-\mu_-^2) = 1-(\frac{1}{2})^2-(1-\frac{1}{2})^2 = \frac{1}{2} \end{aligned}$$

6. I'll justify the errors below that whether they can be written in the form $k(\mu_+-\mu_+^2)$:

[a] $\min(\mu_+,\mu_-) = \min(\mu_+, 1-\mu_+)$, cannot be simplified to $k(\mu_+-\mu_+^2)$.

[b]

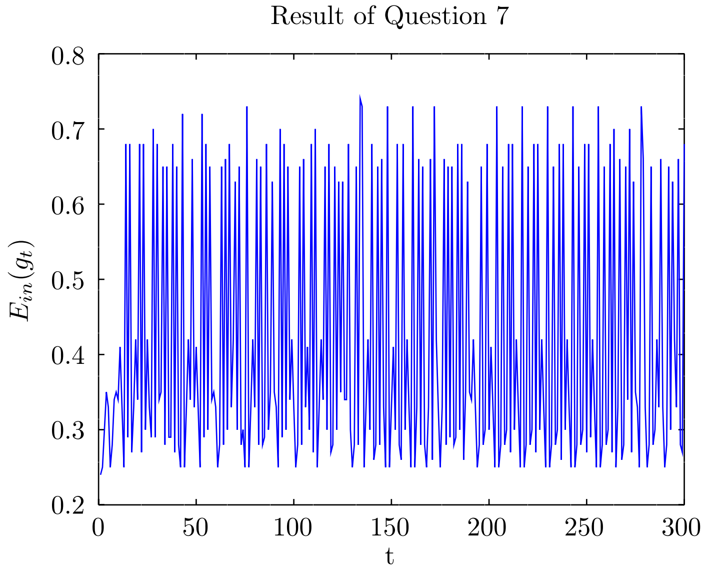
$$\begin{aligned} \mu_+(1-(\mu_+-\mu_-))^2+\mu_-(-1-(\mu_+-\mu_-))^2 &= \mu_+(1-\mu_++\mu_-)^2+\mu_-(-1-\mu_++\mu_-)^2 \\ &= \mu_+(2\mu_-)^2+\mu_-(-2\mu_+)^2 \\ &= 4(\mu_+\mu_-^2+\mu_+^2\mu_-) \\ &= 4(\mu_+(1-\mu_+)^2+\mu_+^2(1-\mu_+)) \\ &= 4(\mu_+-2\mu_+^2+\mu_+^3+\mu_+^2-\mu_+^3) \\ &= 4(\mu_+-\mu_+^2) \\ &= 2\times\text{Gini index} \end{aligned}$$

[c] Apparently the logarithm cannot be simplified to some form of quadratic polynomial.

[d] $1-|\mu_+-\mu_-| = 1-|\mu_+-1+\mu_+| = 1-|2\mu_+-1|$, unlikely to further simplify to $k(\mu_+-\mu_+^2)$.

From above, we conclude that the answer is **[b]** (the squared regression error).

7. The relation between t and $E_{\text{in}}(g_t)$ is shown below.[3] According to the results, $E_{\text{in}}(g_1) = 0.24, \alpha_1 = 0.573640$.

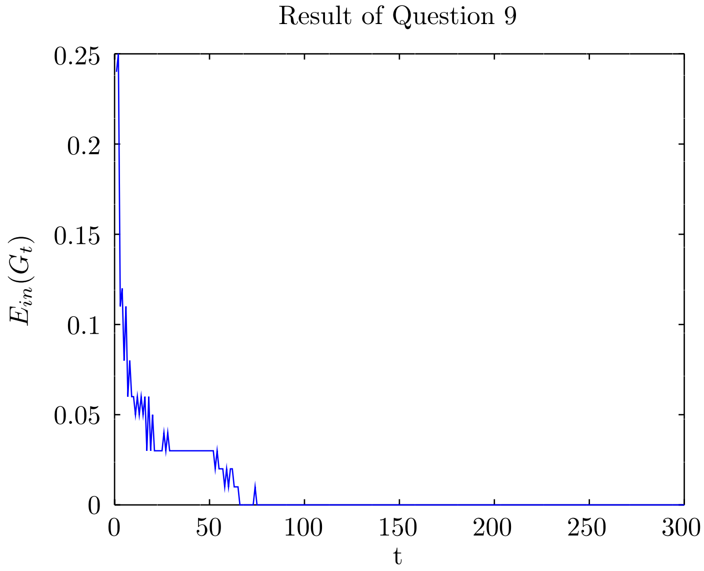


8. Obviously, $E_{\text{in}}(g_t)$ is neither increasing or decreasing. It fluctuates between roughly 0.3 and 0.7, and it seems to have cycles up and down. A Possibility is that after a hypothesis is

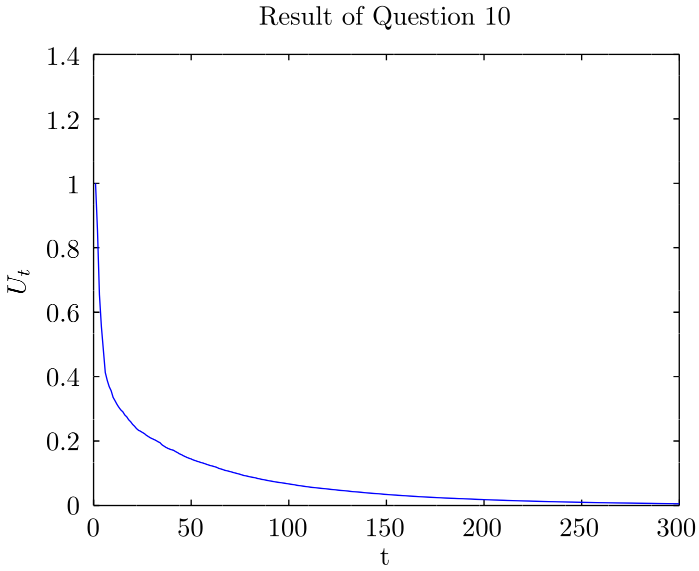
[3] All plots is this answer sheet are made with GNU Octave.

chosen, the wrong data will be emphasized with larger weight and may more likely to be correctly classified in the nest iteration, but the in-sample errors may have a large difference (since the weights are adjusted to make the previous one have the worst performance in terms of weighted in-sample error), which makes the $E_{\text{in}}(g_{t+1})$ when $E_{\text{in}}(g_t)$ is low and $E_{\text{in}}(g_{t+1})$ decrease when $E_{\text{in}}(g_t)$ is high.

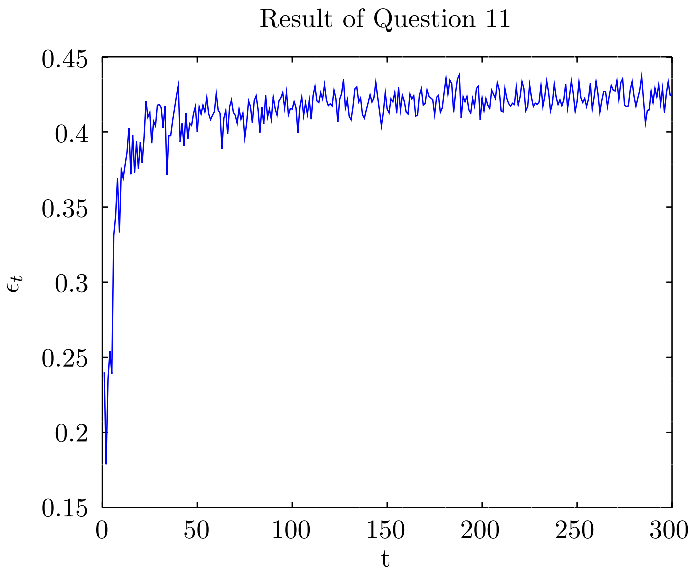
9. The relation between t and $E_{\text{in}}(G_t)$ is shown below. According to the results, $E_{\text{in}}(G) = 0$.



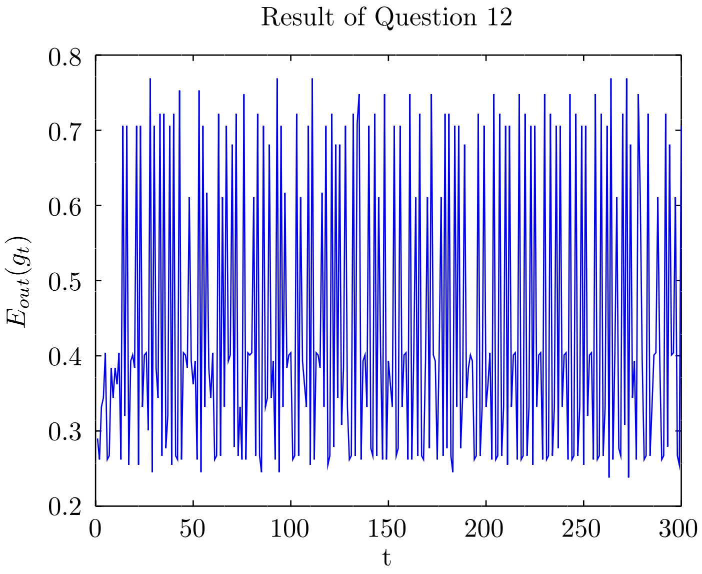
10. The relation between t and U_t is shown below. According to the results, $U_2 = 0.854166, U_T = 0.005465$.



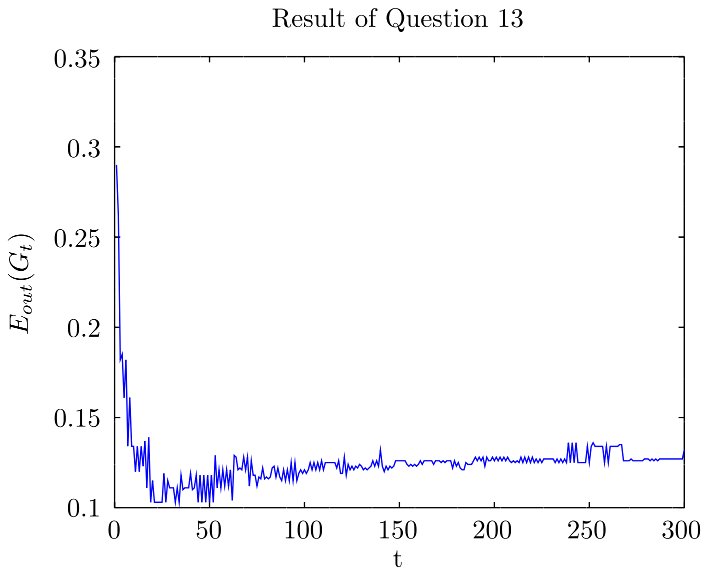
11. The relation between t and ϵ_t is shown below. According to the results, $\min(\epsilon_t) = 0.178728$.



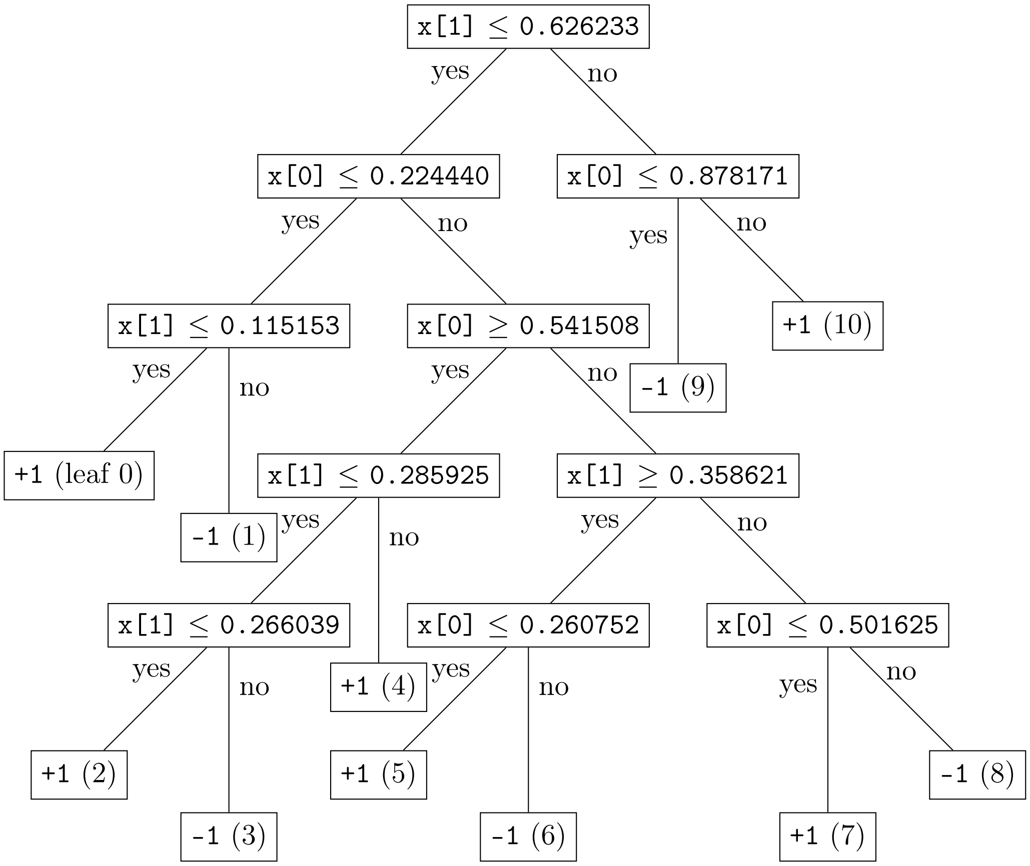
12. The relation between t and $E_{\text{out}}(g_t)$ is shown below. According to the results, $_{\text{out}}(g_1) = 0.29$.



13. The relation between t and $E_{\text{out}}(G_t)$ is shown below. According to the results, $_{\text{out}}(G) = 0.132$.



14. Here is the tree drawn with `tikz`. Each leaf is tagged with a unique number, which will be used in question 16. Moreover, `x[0]` means the 0th feature of input data `x`.



15. According to the result, $E_{\text{in}} = 0$ and $E_{\text{out}} = 0.126$.

16. According to the result, the lowest E_{in} achievable by pruning a leaf is 0.01 (pruning leaf 0, 3, 5 or 8, where the leaf numbers are listed in question 14), and the lowest E_{out} achievable is

0.109 (pruning leaf 8).

17. *Proof.* I'll prove $U_1 = 1$, $U_{t+1} = U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)}$ and $U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \le U_t \cdot 2\sqrt{\epsilon(1-\epsilon)}$ by order.

(a) According to the initial setting of AdaBoost[4], for $n = 1, 2, \cdots, N$, $u_i^{(1)} = \frac{1}{N}$. Thus $U_1 = \sum_{n=1}^{N} u_n^{(1)} = N \times \frac{1}{N} = 1$.

(b) We first define: $U_t^{\mathbf{x}} \equiv \sum_{n=1}^{N} u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]$ (the sum of the weights of incorrect examples). With this, it's easy to justify that $\epsilon_t = \frac{\sum_{n=1}^{N} u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^{N} u_n^{(t)}} = \frac{U_t^{\mathbf{x}}}{U_t}$. According to the algorithm[5], we can get the following:

$$\begin{aligned} U_{t+1} &= U_t^{\mathbf{x}} \cdot \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + (U_t - U_t^{\mathbf{x}}) \cdot \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} \\ &= \epsilon_t U_t \cdot \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + (1-\epsilon_t) U_t \cdot \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} \\ &= U_t \cdot \sqrt{\epsilon_t(1-\epsilon_t)} + U_t \cdot \sqrt{\epsilon_t(1-\epsilon_t)} = U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \end{aligned}$$

(c) Since $\epsilon_t \le \epsilon < \frac{1}{2}$, and since the function $x(1-x)$ monotonically increases in the interval $[0, \frac{1}{2})$, we get $\epsilon_t(1-\epsilon_t) \le \epsilon(1-\epsilon)$ and thus $U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \le U_t \cdot 2\sqrt{\epsilon(1-\epsilon)}$.

Combining the justifications above, the statement is proven. □

18. From the hint, we have $E_{\text{in}}(G_T) \le U_{T+1}$, and from the previous question we get:

$$\begin{aligned} U_{t+1} &= U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \\ &= U_{t-1} \cdot 2\sqrt{\epsilon_{t-1}(1-\epsilon_{t-1})} \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \\ &= \cdots \\ &= U_1 \cdot \prod_{i=1}^{t} (2\sqrt{\epsilon_i(1-\epsilon_i)}) \\ &\le (2\sqrt{\epsilon(1-\epsilon)})^t \end{aligned}$$

Combined together, along with the inequality given in this question (it can be used since $\epsilon < \frac{1}{2}$), we get:

$$E_{\text{in}}(G_T) \le U_{T+1} \le (2\sqrt{\epsilon(1-\epsilon)})^T \le \left(\exp(-2(\frac{1}{2}-\epsilon)^2) \right)^T = \exp(-2T(\frac{1}{2}-\epsilon)^2))$$

Now consider ϵ to be a constant. Consider $T = O(\log N) = k \ln N$, where k is a coefficient. Assign $k = \frac{1}{2(\frac{1}{2}-\epsilon)^2}$, we get:

$$E_{\text{in}}(G_T) \le \cdots \le \exp\left(-2(\frac{1}{2}-\epsilon)^2 \cdot \frac{\ln N}{2(\frac{1}{2}-\epsilon)^2} \right) = \exp(-\ln N) = \frac{1}{N}$$

[4] $\mathbf{u}^{(1)} = [\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N}]$

[5] `http://www.csie.ntu.edu.tw/~htlin/mooc/doc/208_handout.pdf#21`: Step 2

That is $E_{\text{in}}(G_T)$ will be no more than $\frac{1}{N}$ after $O(\log N)$ iterations. Moreover, since $E_{\text{in}}(G_T)$ is actually discrete, which must be a multiple of $\frac{1}{N}$. Now that when the iteration $T = k \ln N + 1$, which is one more iteration than above, and is still $O(\log N)$, the bound of E_{in} will be:

$$E_{\text{in}}(G_T) \leq \exp(-\ln N - 2(\frac{1}{2} - \epsilon)^2) < \frac{1}{N}$$

According to the fact that the value of E_{in} must be at least $\frac{1}{N}$ when it's non-zero, after $T = k \ln N + 1 = O(\log N)$ iterations, $E_{\text{in}}(G_T) = 0$.