EE6550 MACHINE LEARNING

HW#1, REPORT OF MY PROGRAM

102061210 王尊玄

1. Objective:

Implement a consistent PAC-learning algorithm A for the concept class C of all axis-aligned rectangular areas in the plane. Input is given as a random sample $S = (\mathrm{x}_1, \mathrm{x}_2 \dots \mathrm{x}_\mathrm{m})$ of size m drawn i.i.d. according to a fixed but unknown probability distribution P over the input space $\mathbf{R}^2$ with labels $(\mathrm{c}(\mathrm{x}_1), \mathrm{c}(\mathrm{x}_2) \dots \mathrm{c}(\mathrm{x}_\mathrm{m}))$, where c is a fixed but unknown concept. Output is a hypothesis $\mathrm{h}_\mathrm{S} = A(S{:}\, c, H)$, approximating unknown concept c. Let R be generalization error, our goal:

$$\mathrm{P}(\mathrm{R}(\mathrm{h}_\mathrm{S}) < \varepsilon) > 1 - \delta$$

2. Method:

✧ Probability distribution P: We assume that the unknown probability distribution to draw sample is a bivariate normal distribution over input space $\mathbf{R}^2$ with its pdf,

$$\frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-r_{x,y}^2}}e^{-\frac{\frac{\left(x-\mu_x^2\right)^2}{\sigma_x^2}-2r_{x,y}\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}+\frac{\left(y-\mu_y\right)^2}{\sigma_y^2}}{2(1-r_{x,y}^2)}}$$

, where μ_x, σ_x^2 and μ_y, σ_y^2 are mean and variance of x-coordinate and y-coordinate respectively and $r_{x,y} = \frac{E(x-\mu_x)E(y-\mu_y)}{\sigma_x\sigma_y}$ is the correlation coefficient of x-coordinate and y-coordinate. Parameters that can be adjusted are specified,

$$\mathrm{MU} = [\mu_x\ \mu_y]$$

$$\mathrm{SIGMA} = \begin{bmatrix} \sigma_x^2 & r_{xy}\sigma_x\sigma_y \\ r_{xy}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$$

✧ Concept c: Unknown concept c is selected such that $\mathrm{P(c)} \geq 2\varepsilon$, where ε is the upper bound of generalization error guarantee. As it is not realistic to compute $\mathrm{P(c)}$, we use $\hat{p} = \sum_{i=1}^{m} c(x_i)$ as an estimator of $\mathrm{P(c)}$. By central limit theorem, we get with a probability at least 0.9999, we have $\mathrm{P(c)} \geq \hat{p} - \varepsilon$. Thus, if we want to make $\mathrm{P(c)} \geq 2\varepsilon$, then we find c satisfying $\hat{p} > 3\varepsilon$.

✧ PAC-learning algorithm A:

Algorithm A will generate a hypothesis $\mathrm{h_S}$ that encircles all positive points ($\mathrm{c(x)} = 1$), and itself is an axis-aligned rectangle in $\mathbf{R}^2$. Algorithm A will search through all positive points for smallest and largest value of x-coordinate and y-coordinate respectively.

✧ Generalization error $\mathrm{R(h_S)}$:

Given a labeled sample of size $\mathrm{m} = \frac{4}{\varepsilon}\ln\frac{4}{\delta}$, we can get $\mathrm{P(R(h_S) < \varepsilon) > 1 - \delta}$. Since directly compute $\mathrm{R(h_S)}$ is impractical, we use $\hat{q} = \frac{1}{m}\sum_{i=1}^{m}\Delta_S(x_i, c(x_i))$ as an estimator of $\mathrm{P}(\Delta_S) = \mathrm{R(h_S)}$. By central limit theorem, we get with a probability at least 0.9999, we have

$$\hat{q} - \frac{\varepsilon}{10} \le \mathrm{R(h_S)} \le \hat{q} + \frac{\varepsilon}{10}$$

Thus, we can assert that $\mathrm{R(h_S)} < \varepsilon$ if $\hat{q} < \frac{9\varepsilon}{10}$.

Also, if we want to make sure that with a probability at least $1 - \delta$, $\mathrm{R(h_S)} < \varepsilon$ is true, then we run algorithm A $\frac{10}{\delta}$ times to find a unknown but fixed concept c and show that no more than 10 out of $\frac{10}{\delta}$ $\mathrm{h_S}$ defies $\mathrm{R(h_S)} < \varepsilon$.

3. Result:

✧ Input:

MU	r_{xy}	σ_x	σ_y
[5 15]	0.4	0.1	0.15
ε	δ	c	
0.1/0.01	0.01	$\begin{bmatrix} 4.7954 & 5.3170 \\ 14.9089 & 15.3026 \end{bmatrix}$	

✧ Output:

	$\varepsilon = 0.1$	$\varepsilon = 0.01$
$\mathrm{h_S}$	$\begin{bmatrix} 4.8071 & 5.2812 \\ 14.9148 & 15.2852 \end{bmatrix}$	$\begin{bmatrix} 4.8019 & 5.2883 \\ 14.9090 & 15.3026 \end{bmatrix}$
$\hat{q}$	0.0338	0.0021

$$0.0338 < \frac{9}{10} \times 0.1 \rightarrow R(h_S) < \varepsilon = 0.1$$

$$0.0021 < \frac{9}{10} \times 0.01 \rightarrow R(h_S) < \varepsilon = 0.01$$

2694	# of $\mathrm{h_S}$ s.t. $\mathrm{R(h_S)} > \varepsilon$
$\varepsilon = 0.1$	{2, 6, 9, 4, 7, 5, 3, 5, 4, 6}
$\varepsilon = 0.08$	{3, 2, 2, 1, 5, 1, 1, 0, 2, 3}

$\varepsilon = 0.05$	{0, 2, 0, 1, 2, 0, 2, 0, 3, 1}
$\varepsilon = 0.01$	{0, 0, 0, 1, 0, 1, 0, 0, 0, 0}

I took 10 trials over $\frac{10}{\delta}$ times algorithm A, and none of them have more than 10 h_S such that $R(h_S) > \varepsilon$. Also, I find that if we set ε smaller, we can have higher confidence over $R(h_S) > \varepsilon$, i.e. $P(R(h_S) < \varepsilon) > x$, where x may be bigger than $1 - \delta$.