Machine Learning Programming Problem #1

Name: 胡展維
Student ID: 102061247

一.User Manual

Requirement

- Matlab (better r2015a)

How to run

- Simply run function “**HW1_main.m**”

Parameter setup

- Simply set in function **“HW1_main.m**”, as below:

```
%----------------------Parameter setup-----------------------
delta = 0.01;
epsolon = 0.1;
mu = [2, 3];
sigma_x = 1;
sigma_y = 3;
corr_coef = 0.5;
sigma = [sigma_x^2, corr_coef*sigma_x*sigma_y; corr_coef*sigma_x*sigma_y, sigma_y^2];
m = ceil((4/epsolon) * log(4/delta));
%------------------------------------------------------------
```

二.Introduction of my function

- *HW1_main.h*: the main function that set up the parameters and show the result.
- *get_bivariate_normal_distribution.m:* function used to generate the bivariate normal distribution.
- *check_if_larger_3epsolon.m:* function used to find the random concept of the sample size m.
- *find_hs.m:* function used to find out approximated hs(tightest rectangle of).
- *estimate_error.m:* function used to find the error of target concept and our hs.

三.My result

(a) First, I choose correlation coefficient rxy = 0.5, consistent with the problem(a).

(b) For $P(C) \geq 2 * \epsilon$, we must let the empirical probability $\hat{p} = \frac{1}{N} * \sum_{i=1}^{N} c(y_i) \geq 3 * \epsilon$.
The proof in the PDF is as below:

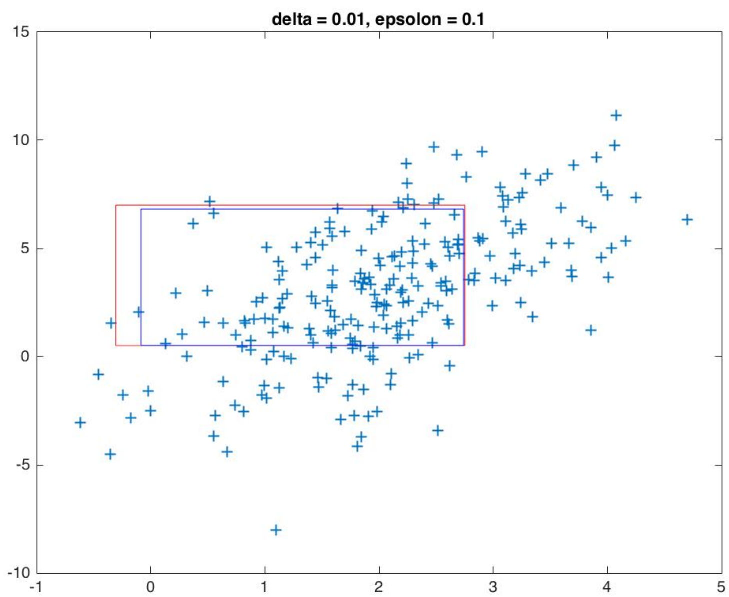
– It is usually difficult, if not impossible, to compute $P(c)$ for given P and c. We resort to an estimation of $P(c)$ as follows. Let $\tilde{S} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ be a sample of size N drawn i.i.d. according to P. Then $c(\mathbf{y}_i)$ is 1 if $\mathbf{y}_i \in c$ and 0 if $\mathbf{y}_i \notin c$. Let $\hat{p} = \frac{1}{N}\sum_{i=1}^{N} c(\mathbf{y}_i)$. It can be seen that $\hat{p}$ is an unbiased estimator of $P(c)$, i.e., $E[\hat{p}] = P(c)$. The variance σ^2 of the estimator $\hat{p}$ is $P(c)(1 - P(c))/N$ which will be very small for large N. Since $0 < P(c) < 1$, it is clear that $\sigma^2 \leq 1/(4N)$. By the central limit theorem, the distribution of $(\hat{p} - P(c))/\sigma$ is well approximated by the standard normal distribution for large enough N. Since $P((\hat{p} - P(c))/\sigma \leq 3.719) \simeq 0.9999$, with probability well approximated by 0.9999, we have $(\hat{p} - P(c))/\sigma \leq 3.719$, i.e., $P(c) \geq \hat{p} - 3.719\sigma$. Since $\sigma \leq 1/(2\sqrt{N})$, with probability at least 0.9999, $P(c) \geq \hat{p} - 1.8595/\sqrt{N}$. Let N_ϵ be an integer such that $1.8595/\sqrt{N_\epsilon} \leq \epsilon$. Then given a sample $\tilde{S}$ of size $N_\epsilon = \lceil (1.8595/\epsilon)^2 \rceil$ drawn i.i.d. according to P, with a probability at least 0.9999, we have $P(c) \geq \hat{p} - \epsilon$. Thus with a confidence at least 0.9999, the probability $P(c)$ of a selected "unknown" concept c will be no less than 2ϵ if the empirical probability $\hat{p} = \frac{1}{N_\epsilon}\sum_{i=1}^{N_\epsilon} c(\mathbf{y}_i)$ is no less than 3ϵ.

If the message shows that "The selected empirical probability of concept c has P_hat >= 3*epsolon !!!", that means we find the probability of the concept $P(C) \geq 2\epsilon$.

(c)

For $\boldsymbol{\delta = 0.01, \epsilon = 0.1}$:

Sample size = 240 points, I get the following result:

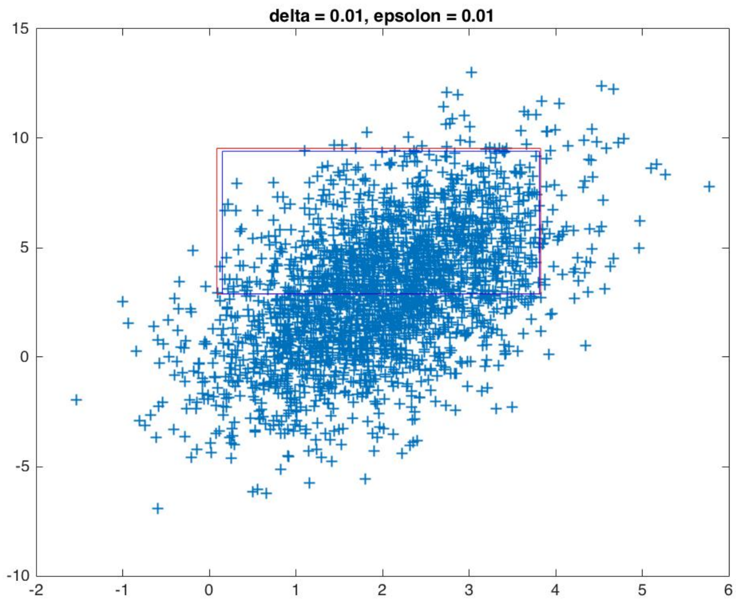


Where the red rectangle is the target concept, and the blue rectangle is our estimated hs. With the empirical error is:

```
The selected empirical probability of concept c has P(c) >= 3*epsolon !!!
The Empirical error is:
    0.0118
```

For $\delta = 0.01, \epsilon = 0.01$:

Sample size = 2397 points, I get the following result:



Where the red rectangle is the target concept, and the blue rectangle is our estimated hs. With the empirical error is:

```
The selected empirical probability of concept c has P(c) >= 3*epsolon !!!
The Empirical error is:
    0.0022
```

(d)

Generalization guarantee: Given a labeled sample S of size $m = \lceil \frac{4}{\epsilon} \ln \frac{4}{\delta} \rceil$, with probability at least $1 - \delta$, the generalization error $R(h_S)$ of the output h_S is upper bounded by ϵ. Your program should be able to verify this guarantee of your PAC-learning algorithm $\mathbb{A}$ by running algorithm $\mathbb{A}$ for $\lceil 10/\delta \rceil$ times and showing that at most 10 out of $\lceil 10/\delta \rceil$, h_S have $R(h_S) > \epsilon$.

For $\delta = 0.01, \epsilon = 0.1$:

The verification of the generalization error is as below:

```
The selected empirical probability of concept c has P(c) >= 3*epsolon !!!
Iteration:
        1000

The Empirical error is:
    0.0181

The number of R(hs) larger that epislon is:
     0
```

The result shows that no R(hs) is larger than ϵ. So we verify generalization guarantee.

<u>For $\boldsymbol{\delta = 0.01, \epsilon = 0.01}$:</u>

The verification of the generalization error is as below:

```
The selected empirical probability of concept c has P(c) >= 3*epsolon !!!
Iteration:
        1000

The Empirical error is:
   9.7907e-04

The number of R(hs) larger that epislon is:
     0
```

The result also shows that no R(hs) is larger than ϵ. So we verify generalization guarantee.