EE6550 MACHINE LEARNING

HW#1, DOCUMENT OF MY PROGRAM

102061210 王尊玄

1. My program is written in Matlab scripts, no need to compile, and can be directly run in Matlab.
2. All test will be done through directly modifying my Matlab code, which is further elaborated in the next point.
3. Any adjustable parameters and the unknown concept will be followed by comment with a lot of plus sign, i.e.
   % ++.
   - ✧ Generalization guarantee parameters,
     δ: delt and ϵ: myEps
   - ✧ Bivariate normal distribution parameters,
     MU: can be directly adjusted through vector variable MU
     SIGMA: can be adjusted through r_xy, sigma_x, sigma_y, where
     $$\text{SIGMA} = \begin{bmatrix} sigma\_x\text{^}2 & r\_xy \times sigma\_x \times sigma\_y \\ r\_xy \times sigma\_x \times sigma\_y & sigma\_y\text{^}2 \end{bmatrix}$$
   - ✧ Unknown concept c can be adjusted directly or randomly
     direct-input: if 1, c can be directly assigned, else, c may be randomly chosen.
     c: there are 2 c (line 24 & 33) followed by comment "directly assigned c" and "randomly chosen c", where random number generator function in "randomly chosen c" must be carefully chosen, otherwise it can never find qualified c and jump out of the searching loop. Also, $\text{c} = [\text{v u}]$, where v and u are both column vector respectively representing lower left corner and upper right corner.
   - ✧ Sample number m is defaulted to $\frac{4}{\epsilon}\ln\frac{4}{\delta}$
4. This program will output a hypothesis $\text{h}_\text{S} = A(S; c, \boldsymbol{H})$, which is found by PAC learning algorithm A over sample S within hypothesis set $\boldsymbol{H}$, equivalent to concept class **C**, as well as an unbiased estimator $\hat{q}$ of $\text{P}(\Delta_\text{S}) = \text{R}(\text{h}_\text{S})$, where our validation goal $\text{R}(\text{h}_\text{S}) < \epsilon$ can be approximated by $\hat{q} \pm 0.1\epsilon < \epsilon$. Finally, a graph will be shown to help visualizing this program.
5. You can run "PAC.m" or "validate.m" for different purposes, but DO comment all the codes creating figures (codes under "figure") in

"PAC.m", otherwise there will be thousands of figures pop out, which will be disastrous. (maybe your GPU memory will be blown up.)

6. Given an unknown but fixed concept c (c must be directly assigned to certain fixed value throughout the following process). A checking program "validate.m" can be executed to validate guarantee $\mathrm{P}(\mathrm{R}(\mathrm{h_S}) < \epsilon) \geq 1 - \delta$ over my PAC learning algorithm A. By running my PAC learning algorithm A ("PAC.m") $\frac{10}{\delta}$ times, at most 10 out of $\frac{10}{\delta}$ $\mathrm{h_S}$ have $\mathrm{R}(\mathrm{h_S}) > \epsilon$. "validate.m" will output "numFail" as the number of $\mathrm{h_S}$ that defies $\mathrm{R}(\mathrm{h_S}) < \epsilon$. If "numFail" is no greater than 10, then "hurrah! We make it!"