Homework #2

1、i) 我们可以简单写出素数规则，即有明确的规则，不需要学习。×

ii) 可能有潜在的规律，即异常点检测问题。✓

iii) 有明确的公式来得出结果。×

iv) 无明确公式可以使用机器学习来解决。✓

v) 无明确规则适宜用机器学习来解决。✓

2、增强学习，首先不能直接分类（非监督），又通过反馈评价每次输出的好坏来改进游戏策略。属于增强学习。

3、无监督，训练集没有明确分类标签。

4、监督，无论是有噪还是无噪，其分类都是明确的。

5、主动学习，算法在训练过程中主动询问，即有选择的训练样本，以期望获得最大信息。

6、$E_{OTS}(g,f) = \frac{1}{L}\sum_{l=1}^{L}[\![g(x_{N+l})\ \text{is even}]\!] = \frac{1}{L}\times\frac{L}{2} = 1/2$.

7、f能拟合D的所有元素，这是训练集内的，所有的f表现也都一样，那么对于后续的L个元素，f可能有不同的拟合效果，即每个样本 $x_{N+L} = \{0, 1\}$ 所以所有f可能数为 2^L。

8、证明如下：（下面的证明来自网上同学的答案）。

$E_f\{E_{OTS}(A(D),f)\}$

$= \frac{1}{2^L}\sum_{i=1}^{2^L}\frac{1}{L}\sum_{k=1}^{L}[[g(x_{N+k}) \ne f_i(x_{N+k})]]$

$= \frac{1}{2^L}\frac{1}{L}\sum_{k=1}^{L}\sum_{i=1}^{2^L}[[g(x_{N+k}) \ne f_i(x_{N+k})]]$

因为前面我们知道 $f_i(x_{N+k})$ 中取值±1各一半. 故对于所有的g其结果都是类似的。

原式 $= \frac{1}{2^L}\frac{1}{L}\sum_{k=1}^{L}2^{L-1} = 1/2$ 此结果与具体的算法无关。

我们可以这么想，由于所有f产生任何D的概率都是一样，即对于测试集没有任何帮助，那么无论我选用哪个g来计算 E_{OTS} 其对于测试集而言其效果是类似的。

9. $\nu=0.5$ 即有5个是橙色弹珠 $P = C_{10}^5 \times 0.5^5 \times (1-0.5)^5 = 0.24$.

10. 同理 $P = C_{10}^9 \times 0.9^9 \times 0.1^1 = 0.39$.

11. 分为两种情况 即 $\nu=0.1$ 和 0, $P = C_{10}^1 \times 0.9^1 \times 0.1^9 + C_{10}^0 \times 0.9^0 \times 0.1^{10}$

12. 由 Hoeffding 不等式得 $P[|\nu-\mu|>\varepsilon] \le 2e^{-2\varepsilon^2 N}$ 取 $\varepsilon=0.8$

代入计算: $P \le 2e^{-2\times 0.64\times 10} = 5.52\times 10^{-6}$

这得到的是一个 upper bound. 因为 ε 取的是0.8

13. 从题意相当各种类型条件的骰子占1/2, 故 $P=(1/2)^5 = 1/32$.

14. 首先解读题意, 题意为从袋子中抽取5个骰子, 总有某1个数字为橙色的概率。

	橙	绿
A	2.4.6	1.3.5
B	1.3.5	2.4.6
C	1.2.3	4.5.6
D	4.5.6	1.2.3

1. B,C
2. A,C
3. B,C
4. A,D
5. B,D
6. A,D

⇒

- B,C 1.3总是橙 ①
- AC 2总是橙 ②
- AD 4.6总是橙 ③
- BD 5总是橙 ④

所以我们取出的要么全是B或C，要么全是A或C，要么全是A或D，要么全是BD，那么统计时全为A或B、C、D的情况会被重复一次（如①②会重复全是C的情况）

故概率为 $(2^5\times4-1^5\times4)\div(4^5)=31/256$.

15、见代码hw01-15.py，收敛时step=45，Last-mistake=135.

16、见代码hw01-15.py 收敛时step均值为40.2，见所作更新次数图，呈近似正态分布，可知随机选择样本对收敛有帮助。

17、见代码hw01-15.py 收敛时step均值为40.3，故可知α的变化并不有助于加速收敛。

18、见代码hw01-18.py 在我机器上average error为0.12，此时的epoch updates为200，当updates为2000时，average error为0.12，并没有改变太多，此时的epoch updates为50。

19、见代码hw01-18.py，当没有选择最优的权重向量时，而是选择更新为50次的最终结果，其均error为0.34，比pocket算法要差。

华南理工大学
SOUTH CHINA UNIVERSITY OF TECHNOLOGY

20. 只修改了 swap 18.时 error 均值为 0.12，与 updates=50 相差不大
但如果把 updates 减小为 30，则 error 均值为 0.15 变化明显.

21. 因此迭代次数在一定范围内如果增加是有助于提升准确率.

21. 没有效果，因为 $T \leq R^2/\rho^2$ 且 $R^2 = \max_n \|x_n\|^2$ $\rho = \min_n y_n \frac{w_f^T}{\|w_f\|} x_n$
故 R^2 与 ρ^2 均是 x_n 的二次表达式，因此也就不会随着 x_n 的变化而变化.