统计学习第五章

贾金柱

Nov 21, 2017

讲课老师

- 主讲人： 贾金柱
  - Email: jzjia@math.pku.edu.cn
- 助教：肖一君
  - Email: xiaoyijun1994@126.com

第五章 Trees and Forest

- 课程目标
  - CART
  - MARS
  - Boosting
  - Random Forests
  - Chap 9，10，15

CART

- Classification and Regression Tree
- piecewise-constant regression
- partion of the feature space

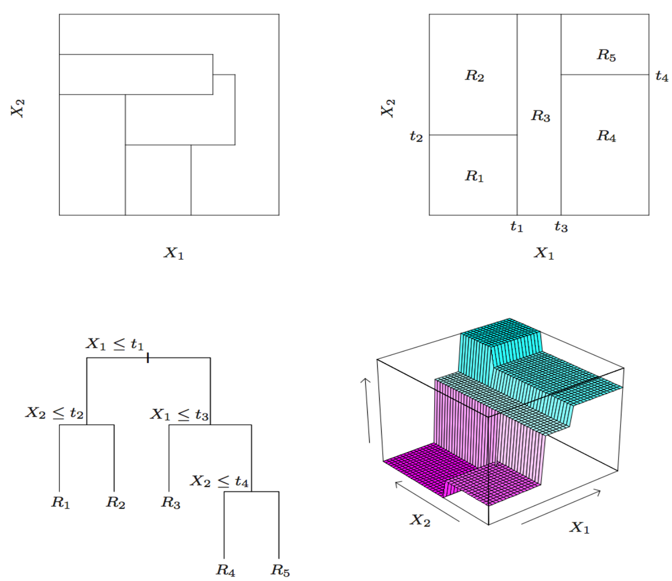

图: piece-wise constant function

Regression Trees

- Input Data: $(x_i, y_i), i = 1, 2, \dots, N$ with $x_i = (x_{i1}, x_{i2}, \dots, x_{ip}) \in \mathbb{R}^p$
- 把数据分成 M 个区域，记为 $R_1, R_2, \dots, R_m$
- Model: $f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$
- 如果 $R_1, R_2, \dots, R_m$ 是已知的, 利用最小二乘 $\min \sum_{i=1}^{n} (y_i - f(x_i))^2$ 可以得到

$$\hat{c}_m = ave(y_i | x_i \in R_m).$$

Regression Trees

- 怎样得到区域的划分?
- greedy method
  - 对于每一变量，可以从某个value处，将空间一分为二

$$R_1(j,s)=\{X|X_j\le s\}\ and\ R_2(j,s)=\{X|X_j>s\}.$$

  - 对于任意的 (j,s) 我们都可以得到一个Loss

$$\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_1)^2$$

  - 寻找 最好的(j,s)，就得到了feature space 的一个划分。
  - 然后再将每一个的划分出来的区域，进行同样的划分
- 什么时候算法停止?

Regression Trees

- 通常的做法是，让 Regression tree 一直长下去， 直至一些叶子结点含有的数据很少（比如5个点）, 形成 T_0
- 然后考虑剪枝
- 定义一些量:

$$N_m = \#\{x_i : x_{\in} R_m\}, \hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

$$C_\alpha(T) = \sum_{m=1}^{T} N_m Q_m(T) + \alpha |T|.$$

Regression Trees

- find the best sub-tree

$$T_\alpha = \arg\min_{T \subset T_0} C_\alpha(T)$$

- choose α by cross-validation and report

$$T_{\hat{\alpha}}$$

Classification Trees

- Regression tree 使用最小二乘来定义loss
- Classification 不能使用最小二乘，需要定义一些新的Loss
- 在节点 m (区域 m), 令

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k).$$

- 令 $k(m) = \arg\max_k \hat{p}_{mk}$

Classification Trees

- 常用的Loss function
  - Misclassification error: $\frac{1}{N_m} \sum_{x_i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$
  - Gini index: $\sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
  - Cross-entroty or deviance: $-\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$

MARS : Multivariace Adaptive Regression Splines

- 一种非线性建模
- 很适合高维数据
- 分片常数（CART） → 分片线性

MARS

- MARS uses expansions in piecewise linear basis functions of the form $(x-t)_+$ 和 $(t-x)_+$

$$(x-t)_+ = \begin{cases} x-t & \textit{if } x > t, \\ 0 & \textit{if } x \leq t \end{cases} \quad \textit{and } (t-x)_+ = \begin{cases} t-x & \textit{if } x < t, \\ 0 & \textit{if } x \geq t \end{cases}$$

- MARS basis

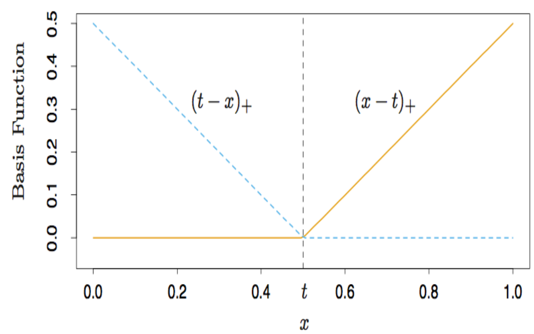

图: Basis for MARS

MARS

- Example:

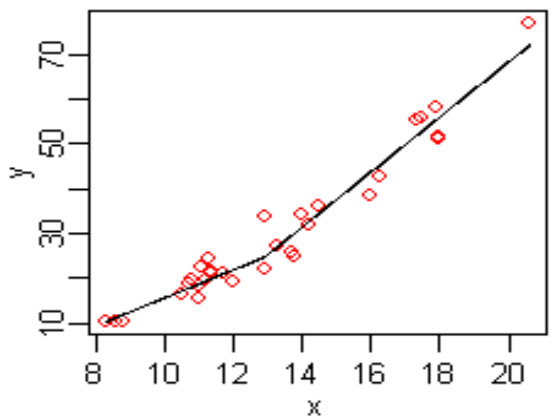

图: Basis for MARS

$$\hat{y} = 25 + 6.1(x - 13)_+ - 3.1(13 - x)_+.$$

MARS

- the model has the form

$$f(x) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(x),$$

  where $h_m(x)$ is a function in $\mathcal{C}$ defined below.

$$\mathcal{C} = \{(X_j - t)_+, (t - X_j)_+\}_{j=1,2,\ldots,n; t \in \{x_{k,j}, k=1,2,\ldots,n\}}.$$

- could add higher-order term.

MARS

- greedy step-wise regression

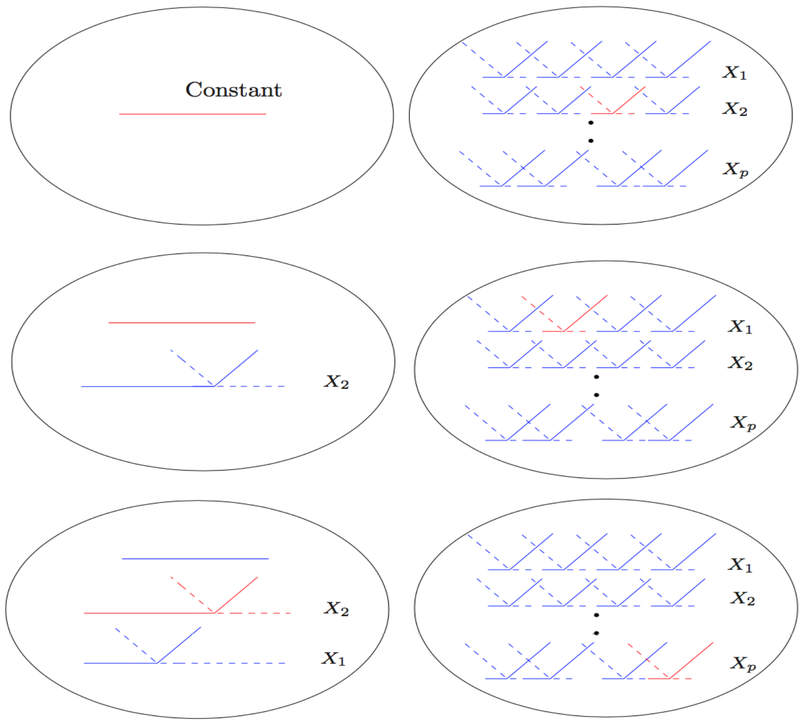

图: Basis for MARS

Boosting

- Boosting 最初是用来做分类问题的
- 它将一些弱分类器组合起来，形成一个能力很强的分类器
- Adaboost
- Logitboost

Boosting

- 考虑二分类问题
- 观测的类别用二值变量 $Y \in \{-1, +1\}$
- 给定预测变量 X, 分类器 $G(x)$ 取值 -1 or +1
- 在训练样本上，分类器的错误率定义为

$$err := \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq G(x_i))$$

- 弱分类器是指那些错误率仅仅比随机猜测好一点的分类器，即

$$err < 0.5$$

Boosting [Discrete adaboost (Freund and Schapire 1996)]

1. 对所有的观测给予同样的权重
2. 重复下面的过程M次:
    1. 使用加权的数据训练一个弱分类器
    2. 计算训练误差 $err = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G(x_i))}{\sum_{i=1}^{N} w_i}$
    3. 计算组合系数 $\alpha_m = \log \frac{1-err_m}{err_m}$
    4. 更新权重 $w_i = w_i \exp[\alpha_m I(y_i \neq G_m(x_i))]$.
3. 组合弱分类器，得到更好的分类器

$$G(x) = sign\Big(\sum_{m=1}^{M} \alpha_m G_m(x)\Big).$$

一个例子： Decision stump

- Decision stump

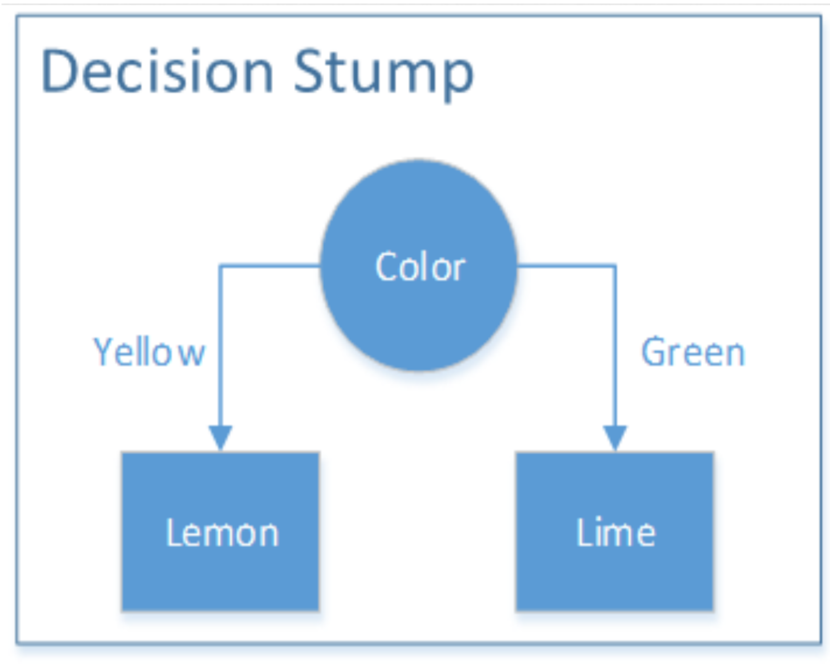

图: decision stump

Decision stump

- 数据结构
  - $X \in \mathbb{R}^{n \times k}$, n 个观测，k个变量
  - $Y \in \{-1, +1\}^n$ lables
  - D – weights
- Stumps – (f, x, s)
  - f: 哪一个变量
  - x: 在哪个地方截断
  - s: +1 or -1
- $\leq (n-1) \times k$ stumps

Decision stump

- 使用 boosting，关键在于如何从加权的data (X，Y， D)训练 week classifier
- 目标函数:maximize information gain or minimize classification error:

$$\min_{j,s} \sum_k D_k(\hat{y}_k(x_j, s) - y_k)^2.$$

Adaboosting

- adaboosting is a newton method to solve an additive logistic regression
- 首先看一个Loss function

$$J(F) = E\left(e^{-yF(x)}\right)$$

引理 $E\left(e^{-yF(x)}\right)$ 的解是：

$$F(x) = \frac{1}{2}\log\frac{P(Y=1|x)}{P(Y=-1|x)}.$$

引理的证明

证明

$$E\left(e^{-yF(x)}\right) = P(y = 1|x)e^{-F(x)} + P(Y = -1|x)e^{F(x)}$$

$$\frac{\partial E\left(e^{-yF(x)}\right)}{\partial F(x)} = -P(y = 1|x)e^{-F(x)} + P(Y = -1|x)e^{F(x)}$$

- 注:

$$P(Y = 1|x) = \frac{e^{F(x)}}{e^{-F(x)} + e^{F(x)}}$$

Adaboosting

定理 Adaboosting 算法本质上是求解 $E(e^{-yF(x)})$ 的newton 法。

证明： 记当前的 $F(x)$ 的估计为 $F(x)$, 现寻求下一步的估计

$$J(F+cf)=E[e^{-y(F(x)+cf(x))}]\approx E\left[e^{-yF(x)}\left(1-ycf(x)+\frac{c^2}{2}f^2(x)\right)\right]$$

上式的≈ 用的是Taylor 展开。

$$\hat{f}(x)=\arg\min_f E\left[e^{-yF(x)}\left(1-ycf(x)+\tfrac{c^2}{2}f^2(x)\right)|x\right]$$

Adaboosting

定义 $E_w(g(x,y)|x) := \frac{E(w(x,y)g(x,y)|x)}{E(w(x,y)|x)}$.

定义 $w(x,y) = e^{-yF(x)}$.

$$
\begin{aligned}
\hat{f} &= \arg\min_f E\Big[e^{-yF(x)}\Big(1 - ycf(x) + \frac{c^2}{2}f^2(x)\Big)|x\Big] \\
&= \arg\min_f E_w[\Big(1 - ycf(x) + \frac{c^2}{2}f^2(x)\Big)|x] \\
&= \arg\min_f E_w[\Big(y - cf\Big)^2|x] \\
&= \arg\min_f E_w[\Big(y - f\Big)^2|x](if \quad c > 0)
\end{aligned}
$$

Adaboosting

上式表明，f 可以通过一个加权最小二乘得到。权重 $w(x,y) = e^{-yF(x)}$

现计算步长 c. 注意

$$
\begin{aligned}
\hat{c} &= \arg\min_c E[e^{-y(F(x)+cf(x))}] \\
&= \arg\min_c E_w[e^{-cyf(x))}] \\
&= \frac{1}{2}\log\frac{1-e}{e},
\end{aligned}
$$

其中 $e = E_w[1_{y \neq f(x)}]$.

Adaboosting

上述步骤给出了如何更新 $F(x)$:

$$F(x) = F(x) + \frac{1}{2}\log\frac{1-e}{e}\hat{f}(x).$$

在下一步Newton迭代步，权重更新为

$$\begin{aligned} w(x,y) &= e^{-y[F(x)+\frac{1}{2}\log\frac{1-e}{e}\hat{f}(x)]} \\ &= e^{-yF(x)} \cdot e^{-[\frac{1}{2}\log\frac{1-e}{e}]y\hat{f}(x)} \\ &= e^{-yF(x)} \cdot e^{-[\frac{1}{2}\log\frac{1-e}{e}](2I_{y\neq\hat{f}(x)}-1)} \\ &\propto w(x,y) \cdot e^{[\frac{1}{2}\log\frac{1-e}{e}]I_{y\neq\hat{f}(x)}} \end{aligned}$$

Real AdaBoost

- AdaBoost 每次循环，定一个 classfier $\hat{f}(x) \in \{-1, +1\}$
- 如果规定，每次循环，$f(x)$ 可以是一般实数，则得到 Real Adaboost 算法

Real AdaBoost

$$J(F(x) + f(x)) = E(e^{-y(F(x)+f(x))}) = E(e^{-yF(x)}e^{-yf(x)})$$

$$\begin{aligned}
\hat{f}(x) &= \arg\min_{f(x)} E(e^{-yF(x)}e^{-yf(x)}|x) \\
&= \arg\min_{f(x)} e^{-f(x)}E(e^{-yF(x)}I_{y=1}|x) + e^{f(x)}E(e^{-yF(x)|x}I_{y=-1}|x) \\
&= \arg\min_{f(x)} e^{-f(x)}E_w(I_{y=1}|x) + e^{f(x)}E_w(I_{y=-1}|x) \\
&= \frac{1}{2}\log\frac{E_w(I_{y=1}|x)}{E_w(I_{y=-1}|x)} \\
&= \frac{1}{2}\log\frac{P_w(Y=1|x)}{P_w(y=-1|x)}
\end{aligned}$$

Real AdaBoost Algorithm

1. 初始化 $w_i = 1/N, i = 1, 2, \ldots, N$

2. Repeat for $m = 1, 2, \ldots, M$:

   (2.1) 估计 $p_m(x) = \hat{P}_w(y = 1|x)$

   (2.2) 计算 $f_m(x) = \frac{1}{2}\log \frac{p_m(x)}{1-p_m(x)}$

   (2.3) 更新weights: $w_i = w_i * e^{-y_i f_m(x_i)}, i = 1, 2, \ldots, N$

3. 输出 Classifier: $sign(\sum_{m=1}^{M} f_m(x))$.

Different Loss functions

- Losses could be viewed as approximations to Misclassification error:

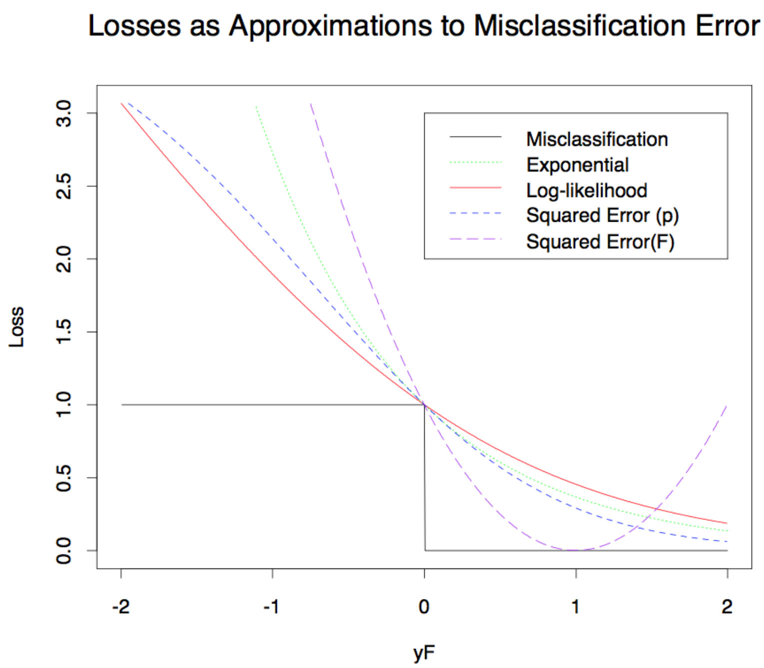

图: Different Loss functions

Log-likelihood Loss functions

$$P(Y = 1|x) = p(x) := \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}.$$

$$P(Y = -1|x) = p(x) := \frac{e^{-F(x)}}{e^{F(x)} + e^{-F(x)}}.$$

$$P(Y = y|x) = \frac{e^{yF(x)}}{e^{yF(x)} + e^{-yF(x)}} = \frac{1}{1 + e^{-2yF(x)}}$$

$$E(\ell(y, x)) := E(\log(P(Y = y|x))) = -E\log(1 + e^{-2yF(x)}).$$

Logit boosting

- If we use expected loglikelihood as the Loss function instead of $E(-yF(x))$, Newton algorithm gives Logit boosting.

考虑从当前$F(x)$ 进一步优化目标函数：

$$E(\ell(F(x)+f(x)))$$

Newton 法告诉我们

$$\hat{f}(x) = -H(x)^{-1}s(x),$$

其中 $s(x) = \frac{\partial E(\ell(F(x)+f(x)))}{\partial f(x)}|_{f(x)=0}$

$H(x) = \frac{\partial^2 E(\ell(F(x)+f(x)))}{\partial f(x)^2}|_{f(x)=0}$

Logit boosting

定义 $p(x) = \frac{e^{F(x)}}{e^{F(x)}+e^{-F(x)}}$

定义 $y^* = (y+1)/2$, 则

$$E(\ell(F(x))) = E(2y^* F(x) - \log(1 + e^{2F(x)}).$$

$$\begin{aligned}
&\frac{\partial E(\ell(F(x)+f(x)))}{\partial f(x)}|_{f(x=0)} \\
&= \frac{-\partial E(2y^*(F(x)+f(x)) - \log(1+e^{2F(x)+2f(x)})}{\partial f(x)}|_{f(x)=0} \\
&= 2E(y^* - p(x)|x)
\end{aligned}$$

$$H(x) = -4E(p(x)(1-p(x))|x).$$

Logit boosting

$$
\begin{aligned}
F(x) &= F(x) - H(x)^{-1}s(x) \\
&= F(x) + \frac{1}{2}\,\frac{E(y^* - p(x)|x)}{E(p(x)(1-p(x))|x)} \\
&= F(x) + \frac{1}{2}E_w\Big(\frac{y^* - p(x)}{p(x)(1-p(x))}|x\Big)\ ,
\end{aligned}
$$

其中 $w(x) = p(x)(1-p(x))$.

注意:

$$
E_w\Big(\frac{y^* - p(x)}{p(x)(1-p(x))}|x\Big) = \arg\min_f E_w\Big(f(x) - \frac{y^* - p(x)}{p(x)(1-p(x))}\Big)^2
$$

Logit boosting Algorithm

1. Start with wieghts $w_i = 1/N, i = 1, 2, \ldots, N$, $F(x) = 0$ and probability estimates $p(x_i) = 1/2$.

2. Reapeat for $m = 1, 2, \ldots, M$:

   (2.1) 计算 working response and weights
   $z_i = \frac{y_i^* - p(x_i)}{p(x_i)(1-p(x_i))}$ $w_i = p(x_i)(1 - p(x_i))$

   (2.2) fit weighted least square
   $f_m = \arg\min_{f \in some\ class} \hat{E}_w \left(f(x) - \frac{y^* - p(x)}{p(x)(1-p(x))}\right)^2$

   (2.3) 更新 $F(x) = F(x) + \frac{1}{2} f_m(x), p(x) = \frac{e^{F(x)}}{e^{-F(x)} + e^{F(x)}}$.

3. Output $sign(F(x)) = sign(\sum_{m=1}^{M} f_m(x))$.

Multi-class

$$p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$$

with $\sum_{k=1}^{J} F_k(x) = 0$.

$$F_j(x) = \log p_j(x) - \frac{1}{J} \sum_{j=1}^{J} \log p_k(x).$$

Random forests

随机森林首先通过 Bootstrap 样本构造一些决策树，然后将这些决策树的结果做一个平均，从而提高预测的精度。随机森林的具体过程如下。

Step 1: For b = 1 to B:

(a)选取Bootstrap 样本

(b)使用选取的样本，如下构造决策树：

随机选取m个变量，使用这m个变量构造回归树或者分类树。

Random forests

Step 2: 得到B颗决策树$\{T_b(x), b = 1, 2, \ldots, B\}$。使用这B颗决策树，综合做最后的估计或预测：

对于回归: $\hat{f}(x) = \sum_{b=1}^{B} T_b(x)$

对于分类：分别使用这B颗分类树做预测，最后使用投票的方式决定预测值。即 $\hat{Y}(x) = \arg\max_k \#\{b : T_b(x) = k\}$.

作业

- Due: Dec 14.

-1. 产生一组simulated data

$$Y = \begin{cases} 1 & if \ \ \sum_{j=1}^{10} X_j^2 > 9.34 \\ -1 & otherwise. \end{cases}$$

使用 stump + Adaboosting 对改组数据分类。Report 你的结果。

-2. 研究 spam email data。（参考书上 10.8）