HW2

1.

令 $P(+1|x) = \alpha$，則如果選 y=+1 當 mini-target 的話，有 1-α的機率會出現 false negative。Cost 是(1-α)*1。如果選 y=-1 當 mini-target 的話，有 t 的機率會出現 false positive。Cost 是α*1000。所以臨界值在於α*1000=1-α，所以$\alpha = \frac{1}{1001}$

2.

根據定義$E_{out}(g) = E_{x\sim P(x)}[g(x) \neq f(x)]$。令總資料量為 D，則$g(x) \neq f(x)$的資料量為 D*$E_{out}(g)$，而$g(x) = f(x)$的資料量為 D*(1-$E_{out}(g)$)。這次的狀況是$P(y = +f(x)|x) = 1 - \epsilon$，所以對於$g(x) \neq f(x)$的資料而言，維持錯誤的只剩$(1 - \epsilon)$*D*$E_{out}(g)$的資料量。變成正確的有$\epsilon$*D*$E_{out}(g)$的資料量。對於$g(x) = f(x)$的資料而言，維持正確的只剩$(1 - \epsilon)$*D*(1-$E_{out}(g)$)的資料量。變成錯誤的有$\epsilon$*D*(1-$E_{out}(g)$)的資料量。所以錯誤情況的期望值是(($1 -\epsilon$)*D*$E_{out}(g)$+ ϵ*D*(1-$E_{out}(g)$))/D=$(1 - \epsilon)$ *$E_{out}(g)$+ ϵ*(1-$E_{out}(g)$)= $E_{out}(g) - 2 * \epsilon * E_{out}(g) + \epsilon$。

3.

硬推。首先$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N}(wx_n - y_n)^2$，所以對於$w$，可以算偏微分$\frac{\partial E_{in}(w)}{\partial w_i} = \frac{1}{N}\sum_{n=1}^{N} 2 * (wx_n - y_n) * x_n$。偏微分等於，所以

$\sum_{n=1}^{N}(wx_n - y_n) * x_n = 0$。展開可得$w\sum_{n=1}^{N} x_n^2 = \sum_{n=1}^{N} y_n x_n$ ，所以 $w = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} x_n^2}$。

4.

沒有給點的數量，這裡我們用積分來模擬點的數量趨近無限的的情況，也更能更精準得出 err，err=$\int_0^1 (w_0 + w_1 x - ax^2 - b)^2 = \int_0^1 w_0^2 + w_1^2 x^2 + a^2 x^4 + b^2 + 2w_0 w_1 x - 2w_0 a x^2 - 2w_0 b - 2w_1 a x^3 - 2w_1 b x - 2abx^2 = w_0^2 + \frac{w_1^2}{3} + \frac{a^2}{5} + b^2 + w_0 w_1 - \frac{2w_0 a}{3} - 2w_0 b - \frac{w_1 a}{2} - w_1 b - \frac{2ab}{3}$。首先對$w_0$做偏微分=0，可得$2w_0 + w_1 = \frac{2a}{3} + 2b$。對$w_1$做偏微分=0，可得$\frac{2w_1}{3} + w_0 = \frac{a}{2} + b$。解聯立可得$w_0 = -\frac{a}{6} + b$，$w_1 = a$。

5.

根據公式，$\mathrm{w_{lin}} = (X^T X)^{-1} X^T y$。其中$\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$。根據題目，對所有$y_n$，有$y_n' = \mathrm{a}y_n + b$，所以$\mathrm{y}' = \begin{bmatrix} \mathrm{a}y_1 + b \\ \mathrm{a}y_2 + b \\ \dots \\ \mathrm{a}y_N + b \end{bmatrix} = \begin{bmatrix} \mathrm{a}y_1 \\ \mathrm{a}y_2 \\ \dots \\ \mathrm{a}y_N \end{bmatrix} + \mathrm{b}\begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} = \mathrm{ay} + \mathrm{b}\begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}$。因此，$\mathrm{w'_{lin}} = (X^T X)^{-1} X^T y' = (X^T X)^{-1} X^T \Bigg(\mathrm{ay} +$

$b\begin{bmatrix}1\\1\\\dots\\1\end{bmatrix}\Bigg) = \mathrm{aw_{lin}} + b(X^TX)^{-1}X^T\begin{bmatrix}1\\1\\\dots\\1\end{bmatrix} = \mathrm{aw_{lin}} + b\begin{bmatrix}1\\0\\\dots\\0\end{bmatrix}$。這裡我們觀察一下$(X^TX)^{-1}X^T\begin{bmatrix}1\\1\\\dots\\1\end{bmatrix}$這個東西。這東西相當於 data 的 y 都是 1 的時候求$\mathrm{w_{lin}}$。因為在這種情況，對所有資料$\mathrm{x_n}$，其第一維度，也就是$\mathrm{x_0}$，都是 1。所有有個非常直觀的解，就是$\mathrm{w_{lin}}$=$\begin{bmatrix}1\\0\\\dots\\0\end{bmatrix}$。這時，對所有$\mathrm{x_n}$ ，$\mathrm{w}_{\mathrm{lin}}^T\mathrm{x_n} = [1,0,0,\dots 0]\begin{bmatrix}1\\\mathrm{x_1}\\\dots\\\mathrm{x_d}\end{bmatrix} = 1$，所以 err=$\left(\mathrm{w}_{\mathrm{lin}}^T\mathrm{x_n} - \mathrm{y}\right)^2$=0。因為是 square error，所以 err 必然大於等於 0。所以$\begin{bmatrix}1\\0\\\dots\\0\end{bmatrix}$必然是最優解，因為不可能有 err 比 err=0 還小。題目說$\mathrm{w_{lin}}$有唯一性，所以$\begin{bmatrix}1\\0\\\dots\\0\end{bmatrix}$是唯一的，所以符合$(X^TX)^{-1}X^T\begin{bmatrix}1\\1\\\dots\\1\end{bmatrix} = \begin{bmatrix}1\\0\\\dots\\0\end{bmatrix}$。因此$\mathrm{aw_{lin}} +$

$$b(X^TX)^{-1}X^T\begin{bmatrix}1\\1\\\dots\\1\end{bmatrix} = \mathrm{aw_{lin}} + b\begin{bmatrix}1\\0\\\dots\\0\end{bmatrix}$$。

6.

根據定義，$\mathrm{E_{in}}(w) = \frac{1}{N}\sum_{n=1}^{N}\ln(1+\exp(-y_n w^T x_n))$。對於任意的$\mathrm{w_i}$，上課有說過，$\frac{\partial \mathrm{E_{in}}(w)}{\partial \mathrm{w_i}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+\exp(y_n w^T x_n)}(-y_n x_{ni}) = \frac{1}{N}\sum_{n=1}^{N} h_t(y_n x_n)\,(-y_n x_{ni})$。這時，任意再挑一個$\mathrm{w_j}$，則$\frac{\partial^2 \mathrm{E_{in}}(w)}{\partial \mathrm{w_i}\partial \mathrm{w_j}} = \frac{1}{N}\sum_{n=1}^{N}\frac{y_n x_{ni}}{((1+\exp(y_n w^T x_n))^2} * \exp(y_n w^T x_n) * y_n x_{nj} = \frac{1}{N}\sum_{n=1}^{N}\frac{x_{ni}x_{nj}}{((1+\exp(y_n w^T x_n))^2} * \exp(y_n w^T x_n)$。$\frac{1}{N}\sum_{n=1}^{N}\frac{x_{ni}x_{nj}}{((1+\exp(y_n w^T x_n))^2} * \exp(y_n w^T x_n)$就是$\mathrm{A_E}(w)$的第 ij 項。所以當$X$=$\begin{bmatrix}\mathrm{x_0}\\ \mathrm{x_1}\\ \cdots\\ \mathrm{x_N}\end{bmatrix}$時(X 是個 N*(d+1)矩陣)，$\mathrm{X^T}X$的第 ij 項就是$\sum_{n=1}^{N} x_{ni}x_{nj}$。給定$\mathrm{w_t}$，$\mathrm{A_E}(\mathrm{w_t})$可以表示為$\mathrm{X^T DX}$，因為D是 diagonal，所以 D 的對角線上第 i 個值是$\frac{1}{\mathrm{N}}\frac{\exp(y_i w_t^T x_i)}{((1+\exp(y_i w_t^T x_i))^2} = \frac{1}{N}\frac{1}{1+\exp(y_i w_t^T x_i)} * \frac{\exp(y_i w_t^T x_i)}{1+\exp(y_i w_t^T x_i)} = \frac{1}{N}\frac{1}{1+\exp(y_i w_t^T x_i)} * \frac{1}{1+\exp(-y_i w_t^T x_i)} = \frac{1}{N}h_t(y_i x_i) * h_t(-y_i x_i)$。I 總共有 N 個，所以 D 是個 N*N 矩陣

7.

PLA 就是每一步挑錯誤的 1 個點，做$\mathrm{w_{t+1}} = w_t + y_n x_n$。題目要求的演算法是每一步挑 1 個點，做$\mathrm{w_{t+1}} = w_t - \eta\nabla err = w_t - \eta\nabla\max(0,1-yw^T x)$ 。而$\max(0,1-yw^T x)$就是$\begin{cases} 0 \ \ if \ \ yw^T x \geq 1 \\ 1-yw^T x \ \ if \ \ yw^T x < 1\end{cases}$，所以對任意$w_\mathrm{i}$，偏微分是

$\begin{cases} 0 \;\; if \;\; yw^Tx \geq 1 \\ -yx_i \;\; if \;\; yw^Tx < 1 \end{cases}$，所以梯度是$\begin{cases} 0 \;\; if \;\; yw^Tx \geq 1 \\ -yx \;\; if \;\; yw^Tx < 1 \end{cases}$，也就是 $\max(0,1-yx)$。所以更新$\mathrm{w_{t+1}}$的算式就是$\mathrm{w_{t+1}} = w_t + \eta y_n x_n$ if $yw^Tx < 1$。兩者比較，PLA 是yw^Tx<0 的點才會更新。SGD 是 yw^Tx<1 的點就會更新。條件比較寬，所以 SGD 可以停下來，PLA 就可以停下來。但 PLA 可以停下來的，SGD 不一定可以，可能所有 data 都符合$0 < yw^Tx < 1$，這樣 SGD 就停不下來。根據這個式子，可以發現 x 的大小會對 SGD 的判斷產生影響，所以 normalized 會對 SGD 能不能停下有影響。而且 SGD 有η這個參數，會決定 SGD 會跑多快。

8.

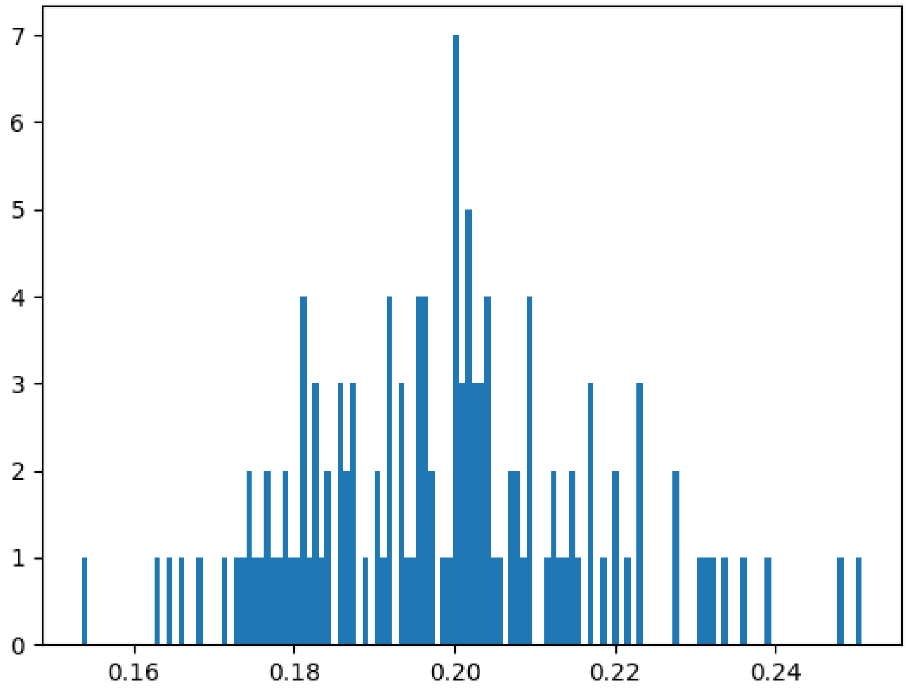
$\mathrm{E_{in}}(W) = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}\ln\frac{\exp(w_{y_n}^T x_n)}{\sum_{i=1}^{K}\exp(w_i^T x_n)} = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}(\ln\exp\left(w_{y_n}^T x_n\right) - \ln(\sum_{i=1}^{K}\exp(w_i^T x_n))) = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}(w_{y_n}^T x_n - \ln(\sum_{i=1}^{K}\exp(w_i^T x_n)))$。對於 W 中任意的$\mathrm{w_{kj}}$，做偏微分。當$y_n = k$時，$\mathrm{w_{kj}}$的偏微分如下。

$\frac{\partial \mathrm{E_{in}}(W)}{\partial \mathrm{w_{kj}}} = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}(x_{nj} - \frac{x_{nj}\exp\left(w_k^T x_n\right)}{\sum_{i=1}^{K}\exp\left(w_i^T x_n\right)})) = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}x_{nj}(1 - \frac{\exp\left(w_{y_n}^T x_n\right)}{\sum_{i=1}^{K}\exp\left(w_i^T x_n\right)})) = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}x_{nj}(1 - \mathrm{h_{y_n}}(x_n)))$。當$y_n \neq k$時，$\mathrm{w_{kj}}$的偏微分如下。$\frac{\partial \mathrm{E_{in}}(W)}{\partial \mathrm{w_{kj}}} = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}(-\frac{x_{nj}\exp\left(w_k^T x_n\right)}{\sum_{i=1}^{K}\exp\left(w_i^T x_n\right)})) = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}x_{nj}(-\frac{\exp\left(w_{y_n}^T x_n\right)}{\sum_{i=1}^{K}\exp\left(w_i^T x_n\right)})) = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}x_{nj}(-\mathrm{h_{y_n}}(x_n)))$。當$y_n = k$時，集合$\mathrm{w_{kj}}$所有 j，則$\nabla \mathrm{w}_{y_n} = \frac{-1}{\mathrm{N}}\sum_{n=1}^{N}x_n(1 - \mathrm{h_{y_n}}(x_n)))$，是個 (d+1)*1 的矩陣。當$y_n \neq k$時，集合$\mathrm{w_{kj}}$所有 j，則$\nabla \mathrm{w_k} =$

$\frac{-1}{\mathrm{N}}\sum_{n=1}^{N} x_n(-\mathrm{h}_k(x_n)))$，是個(d+1)*1 的矩陣。把$\nabla \mathrm{w}_1 \sim \nabla \mathrm{w}_\mathrm{K}$這 K 個向量橫著排起來，就是一個(d+1)*K 的矩陣，也就是$\nabla \mathrm{E}_{\mathrm{in}}(W)$。

9.

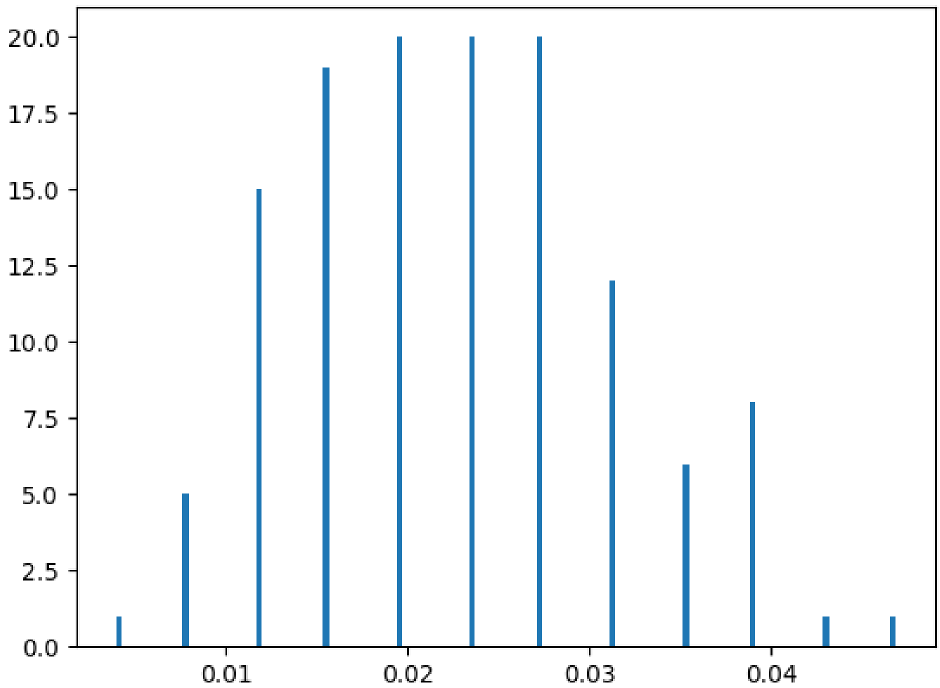
這是我畫的圖:



err 中位數=0.199

10

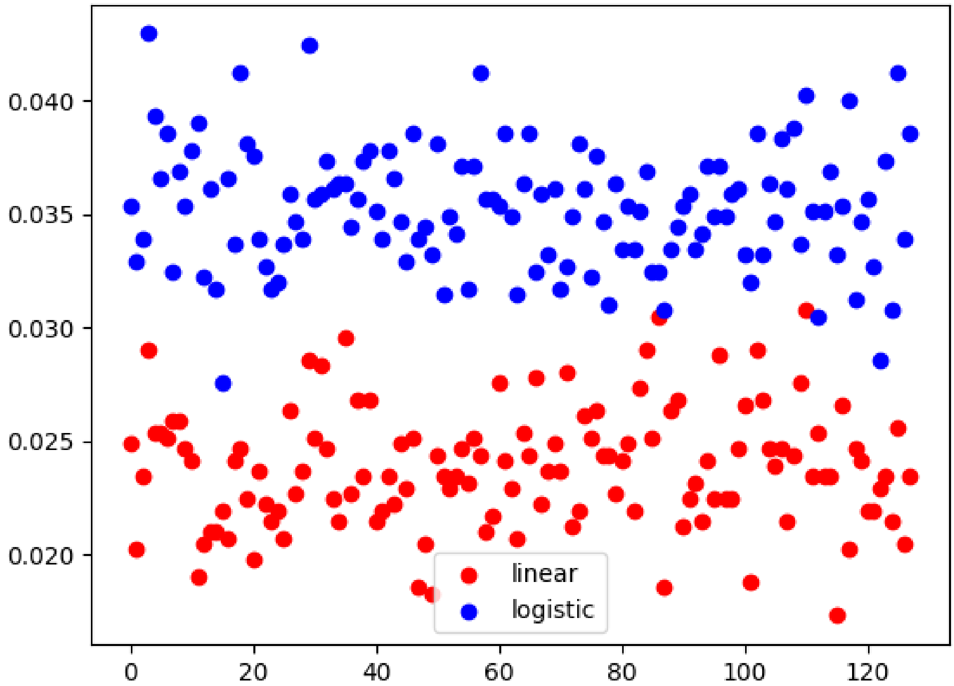
這是我畫的圖:



Err 中位數是 0.023。也就是 256 個平均約 5 個出錯。如上課所說 sqr err 比 0/1 err 大。

11.

這是我畫的圖:

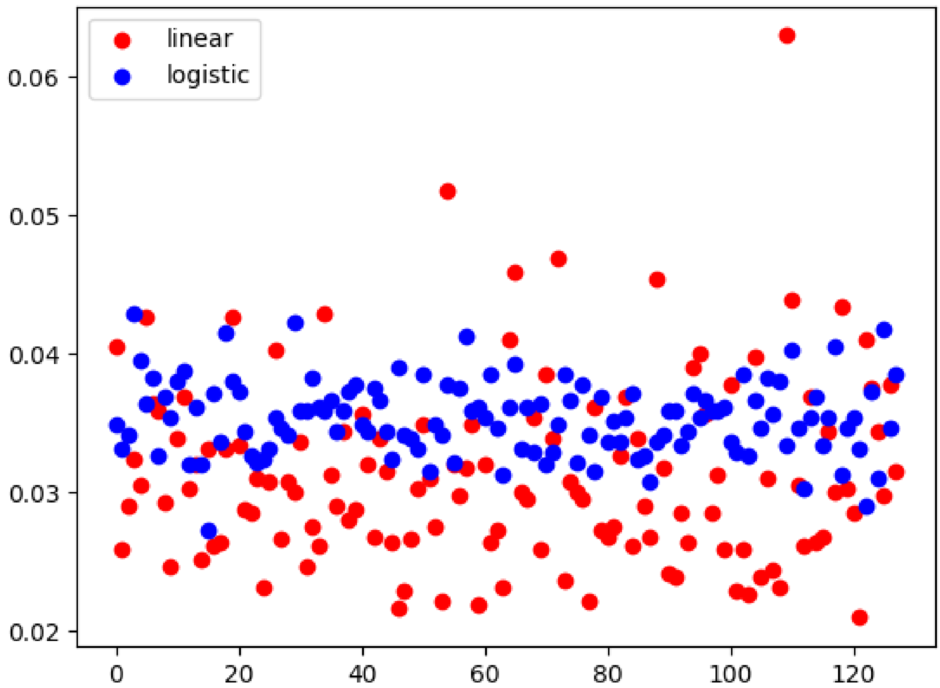


Linear regression 的 median err 是 0.023

Logistic regression 的 median err 是 0.034

12.

這是我畫的圖:



Linear regression 的 median err 是 0.030

Logistic regression 的 median err 是 0.035

linear regression 的 err 變化幅度較大，而 Logistic regression 的 err 變化幅度較小。

可以發現說 linear regression 比較受雜訊的影響，而 Logistic regression 受到的影響比較小。

13.

對於$\hat{X}$而言，因為我第 6 題算出來的 D 其對角第 i 項是 $\frac{1}{N}h_t(y_i x_i) * h_t(-y_i x_i)$，把這個值叫做$D_i$好了。只要把每個$D_i$開根號，乘到 X 裡面每第 i 個 row 就好了。因為h_t必為正，所以D_i開根

號必為實數。

對於$\hat{y}$而言，考慮第 6 題的$\mathrm{E_{in}}(w)=\frac{1}{N}\sum_{n=1}^{N}\ln(1+\exp(-y_n w^T x_n))$，

$\frac{\partial \mathrm{E_{in}}(w)}{\partial \mathrm{w_i}}=\frac{1}{N}\sum_{n=1}^{N}\frac{1}{1+\exp(y_n w^T x_n)}(-y_n x_{ni})=\frac{1}{N}\sum_{n=1}^{N}h_t(y_n x_n)(-y_n x_{ni})$，所以

$-\nabla \mathrm{E_{in}}(w_t)=\frac{-1}{N}\sum_{n=1}^{N}h_t(y_n x_n)(-y_n x_n)=X^T *$

$$\begin{bmatrix}\frac{1}{N}h_t(y_1x_1)(y_1)\\ \frac{1}{N}h_t(y_2x_2)(y_2)\\ \cdots\\ \frac{1}{N}h_t(y_Nx_N)(y_N)\end{bmatrix}=\hat{X}^T\cdot\begin{bmatrix}\frac{\frac{1}{N}h_t(y_1x_1)(y_1)}{sqrt\left(\frac{1}{N}h_t(y_1x_1)*h_t(-y_1x_1)\right)}\\ \frac{\frac{1}{N}h_t(y_2x_2)(y_2)}{sqrt\left(\frac{1}{N}h_t(y_2x_2)*h_t(-y_2x_2)\right)}\\ \cdots\\ \frac{\frac{1}{N}h_t(y_Nx_N)(y_N)}{sqrt\left(\frac{1}{N}h_t(y_Nx_N)*h_t(-y_Nx_N)\right)}\end{bmatrix}=\hat{X}^T\cdot$$

$\begin{bmatrix}sqrt(\frac{h_t(y_1x_1)}{h_t(-y_1x_1)})(y_1)\\ sqrt(\frac{h_t(y_2x_2)}{h_t(-y_2x_2)})(y_2)\\ \cdots\\ sqrt(\frac{h_t(y_Nx_N)}{h_t(-y_Nx_N)})(y_N)\end{bmatrix}$。因此$\hat{y}=\begin{bmatrix}sqrt(\frac{h_t(y_1x_1)}{h_t(-y_1x_1)})(y_1)\\ sqrt(\frac{h_t(y_2x_2)}{h_t(-y_2x_2)})(y_2)\\ \cdots\\ sqrt(\frac{h_t(y_Nx_N)}{h_t(-y_Nx_N)})(y_N)\end{bmatrix}$