# 九、Linear Regression 线性回归

## 9.1 Linear Regression Problem 线型回归问题

在第二章中提到的银行发放信用卡问题，通过是否发放信用卡引出了二元分类问题；本章再次使用这个例子通过发放用户多大额度的信用卡引出回归（regression）或者说线性回归（linear regression）的问题。回归问题与二元分类问题最大的不同在于输出空间，二元分类的输出空间为二元标记，要么+1要么-1，而回归问题的输出空间是整个实数空间，即$y \in R$。

以银行发放信用卡为例，输入集合依然是用户的特征空间，如年龄，收入等等，可以使用与二元分类一致的表示方式$x^T = [X_0, X_1, \cdots, X_d]$；因为输出集合的转变导致回归问题的假设函数与二元分类中的有所不同，但思想一致，仍需考虑对每个输入样本的各个分量进行加权求和，因此最终目标函数f（含有噪音，使用y表示）的表示如公式9-1所示。

$$y \approx \sum_{i=0}^{d} W_i X_i$$

而假设函数的向量表示如公式9-2所示：

$$h(x) = w^T X \qquad \text{公式 9-2}$$

从公式9-2的表示可以方便看书，与二元分类假设函数的表示只差了一个取正负号的函数sign。

使用图像的方式更形象的描述线性回归，如图9-1所示。

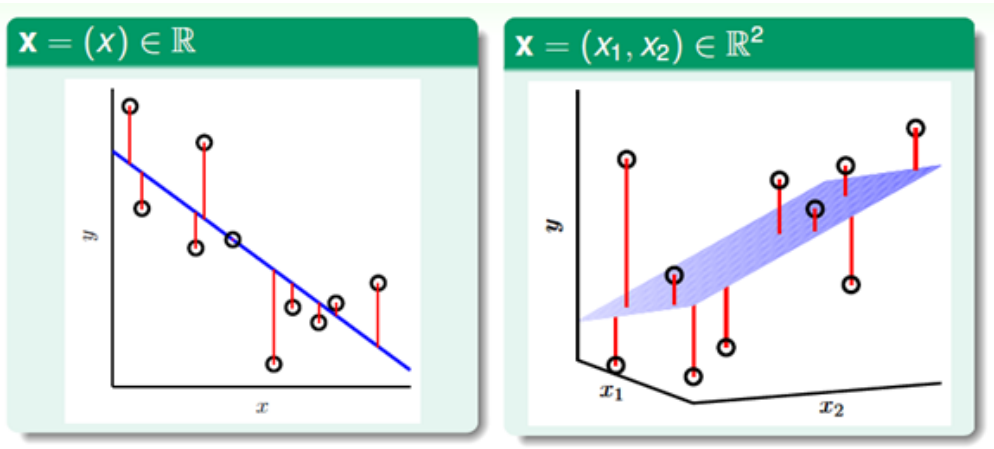


图9-1 a) 1维输入空间的线性回归 b) 2维空间的线性回归

图9-1a中表示输入空间为1维的线性回归表示，其中圆圈○表示输入样本点，蓝色直线表示假设函数$h(x) = W^T X$，连接圆圈与蓝色直线之间的红色线段表示样本点到假设函数的距离，称为剩余误差（residuals），在9-1b中有类似表示。而设计算法的核心思想是使总体剩余误差最小。

上一章中也提到过回归使用的错误衡量是平方误差$err(\vec{y}, y) = (\vec{y}, y)^2$，因此$E_{in}(h)$如公式9-3所示。

$$E_{in}(h) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2 \qquad \text{公式 9-3}$$

同理，$E_{out}(W)$表示公式9-5所示，注意这里使用的是含有噪音的形式，因此$(x, y)$服从联合概率分布P。

$$E_{out}(W) = E_{(x,y)\sim P}(W^T x_n - y_n)^2 \qquad \text{公式 9-5}$$

VC限制可以约束各种情况的学习模型，当然回归类型的模型也被也受此约束，想要学习到知识，只需要寻找$E_{in}(W)$足够小可以满足$E_{out}(W)$足够小的需求。

# 9.2 Linear Regression Algorithm 线性回归算法

此节重点是如何寻找最小的$E_{in}(W)$，为了表达的方便，将求和公式转换成向量与矩阵的形式，将公式9-4转换成公式9-6的形式。

$$E_{in}(W) = \frac{1}{N}\sum_{n=1}^{N}(W^T x_n - y_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n^T W - y_n)^2$$

(次方便显示将 向量W与向量X位置交换，因为是向量内积，符合交换律)

$$= \frac{1}{N}\left\| \begin{matrix} x_1^T W - y_1 \\ x_2^T W - y_2 \\ \cdots \\ x_N^T W - y_N \end{matrix} \right\|^2$$

（将平方和转换成矩阵平方的形式）

$$= \frac{1}{N}\left\| \begin{bmatrix} -x_1^T- \\ -x_2^T- \\ \cdots \\ -x_N^T- \end{bmatrix} W - \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix} \right\|^2$$

（再拆解成矩阵X和向量w与向量y的形式）

$$= \frac{1}{N}\left\| \underbrace{X^R}_{N\times(d+1)} \underbrace{w}_{(d+1)\times 1} - \underbrace{y}_{N\times 1} \right\|^2$$

再回到最初的目标寻找一个最小的$E_{in}(W)$， 如公式9-7所示：

$$\min_w E_{in}(W) = \min_w ||X^T w - y||^2$$

求解此问题，需要了解左式，其唯一（d=1时）示意如图9-2所示

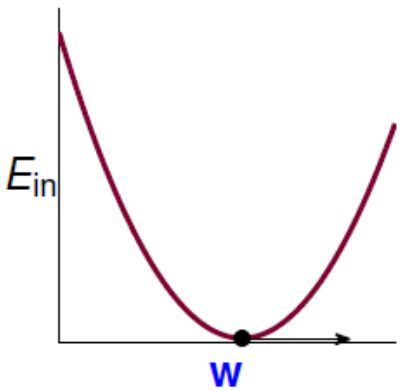


图9-2 一维示意图

可以看出该函数为连续（continuous）、可微（differentiable）的凸（convex）函数，其中连续及可微的概念，学过高等数学的都应该有所了解，凸函数说的通俗点就如图9-2所示，像一个山谷一样的形式（注意国内数学教材中的凹函数是这里凸函数的定义，有点囧），寻找的最佳的$E_{in}(W)$便是山谷的最低点， 对应图中的黑点，以数学的形式表示即梯度（gradient）为0 的点。（我理解的梯度，大概意思是某一向量其各个分量的偏导数组成的向量），梯度为的表示方式如公式9-8所示。

$$\nabla_w E_{in}(W) = \begin{bmatrix} \frac{\partial E_{in}(w)}{\partial w_1} \\ \frac{\partial E_{in}(w)}{\partial w_2} \\ \vdots \\ \frac{\partial E_{in}(w)}{\partial w_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

其中 $\nabla$ 即梯度符号，需要寻找的是$W_{LIN}$，该向量满足$\nabla_w E_{in}(W_{LIN}) = 0$，这里$W_{LIN}$的下标表示线型linear的意思，紧接着的问题是如何求解$\nabla_w E_{in}(W_{LIN}) = 0$ 的$W_{LIN}$。

继续对公式9-6做转化，如公式9-9所示。

$$E_{in}(w) = \frac{1}{N}||X^T w - y||^2 = \frac{1}{N}(w^T \underbrace{X^T X}_{A} w - 2w^T \underbrace{X^T y}_{b} + \underbrace{y^T y}_{c})$$

其中$X^T X$用矩阵A表示，$X^T y$用向量b表示，$y^T y$用标量c表示，紧接着对$E_{in}(w)$求梯度。向量对向量求导，可能很多人都没有接触甚至没有听说过，最多也就是了解向量对某标量求导。可通过图$9-3$在$w$为标量情况下的对比形式，理解E_{in}(w)$求梯度的步骤。

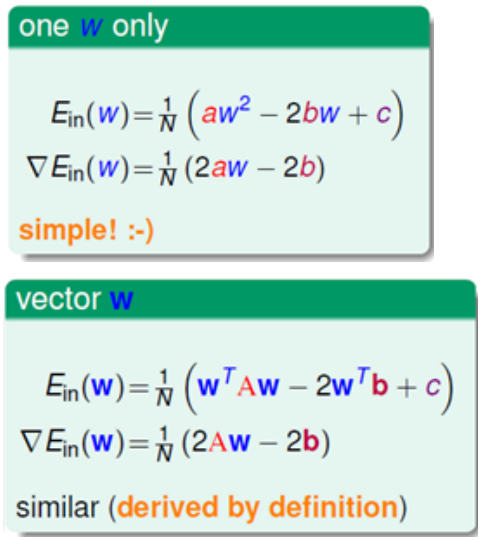


图9-3 a) w为标量时求$E_{in}(W)$的梯度，b)w为向量时$E_{in}(w)$的梯度

线性代数的美妙之处就在于此，如此的相似，因此$\nabla E_{in}(w)$ 可以写成公式9-10的形式

$$\nabla_w E_{in}(W) = \frac{2}{N}(X^T X w - X^T y)$$

另公式9-10求梯度结果为0，即使$E_{in}(w)$ 最小。在输入空间X与输出向量y都为已知的情况下，如何求解最佳的假设函数，$W_{LIN}$ 呢？求解该问题分为两种情况，一是在$X^T X$可逆的情况下，求解该问题很简单，将公式9-10右边的部分设为0，如公式9-11所示。

$$W_{LIN} = \underbrace{(X^T X)^{-1} X^T}_{X^\dagger} y \quad \text{公式 9-11}$$

其中$X^\dagger$ 表示矩阵X的伪逆（pseudo-inverse），注意此处输入矩阵X在很少的情况下才是方阵（N=d+1时）。而这种伪逆矩阵的形式和方阵中的逆矩阵具有很多相似的性质，因此才有此名称。还有一点需要说明，$X^T X$大部分的情况下是可逆的，原因是在进行机器学习时，通常满足$N \geq d+1$，即样本数量N远远大于样本的维度d加1，因此在$X^T X$中存在足够的自由度使其可以满足可逆的条件。

另一种是$X^T X$不可逆的情况，实际上可以得到许多满足条件的解，只需要通过其他的方式求解出$X^\dagger$，选择其中一个满足条件$W_{LIN} = X^\dagger y$的解。

总结下线性回归算法的求解过程，首先通过已知的数据集，构建输入矩阵X与输出向量y，如公式9-12所示。

$$X^T = \underbrace{\begin{bmatrix} -x_1^T- \\ -x_2^T- \\ \vdots \\ -x_N^T- \end{bmatrix}}_{N\times(d+1)} \qquad y = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{N\times 1}$$

通过公式9-12直接求得违逆 $\underbrace{X^{\dagger}}_{(d+1)\times N}$

再通过公式9-11求得假设函数，如公式9-13所示

$$\underbrace{W_{LIN}}_{(d+1)\times 1} = X^{\dagger}y$$

# 9.3 Generalization Issue 泛化问题

本小节讨论的问题理解起来不简单，目前自己还是一知半解，如有表述不正确的地方还希望指正。

首先要回答一个问题，上一小节中使用到的求解最佳假设函数$W_{LIN}$的算法，是否能算是机器学习？

如回答不是，其理由很简单，求解只一步就完成了，不像前面章节中提到的学习方法需要很多步的过程。实际上，这种求解方式在数学中被称作解析解（analytical solution）或者叫封闭解或闭式解（closed-form solution），此种解是一些严格的公式，给出任意的自变量就可以求出其因变量，通常与数值解对应。因此这种求解方式并不像之前提到的PLA等算法时一步一步迭代求出的$E_{in}(W)$的最小解。

回答是的理由更看重结果，这种直接求解方式是数学推导中的精确解，因此求出的$W_{LIN}$一定是$E_{in}(w)$的最小解，符合求解条件，而且求解伪逆算法（此方法被称为高斯消元法，又见高斯，查了一下一共有110项以他名字命名的成果，整个机器学习笔记中你还会不断的听到以他命名的成果）并非如公式展示中显示的那样，一步就可以得出最终结果，而是需要几次的循环迭代（观察了矩阵求伪逆的程序，好像是三层循环，也就印证了NG在他机器学习课程中提到的矩阵求逆的复杂度为$O(N^3)$,只是被程序封装的看不出迭代的过程而已。而判断是否发生机器学习过程最主要标准是学习到的$E_{out}(w)$足够好！

其实通过改进VC限制，也可以证明在线性回归问题中VC起到了很好的约束作用，即找到了好的$E_{in}(W)$就可以保证$E_{out}(W)$还不错，这里不再证明，因为是件非常繁琐的过程。此处只需要记住VC限制不只在二元分类问题中起作用，在线性回归问题中也发挥着作用。

但是本节使用一种比VC限制更容易证明的保证，来说明解析解也可以得到一个好的$E_{out}(W)$

以下给出证明：为什么解析解求出的$E_{in}(W_{LIN})$的结果是好的。而有关的$E_{out}(W_{LIN})$证明与之类似。

首先观察$E_{in}$ 的平均，用符号$\bar{E}_{in}$ 表示，可写成公式9-14所示。

$$E_{in} = E_{D\sim P^N}\{E_{in}(W_{LIN}wrtD)\} = noiselevel\cdot(1-\frac{d+1}{N})$$

其中E表示期望，不断的从整体样本空间中抽取样本集，算其平均值，w.r.t表示关于，noise level 表示数据中的噪音，N为每次抽样的样本数量，d+1为权值向量w的纬度。

从上一节中得知，可以将$E_{in}(W_{LIN})$ 写成公式9-15， 注意y与$\vec{y}$ 是向量形式。

$$E_{in}(W_{LIN}) = \frac{1}{N}||y-\vec{y}||^2 = \frac{1}{N}||y-\underbrace{XX^{\dagger}y}_{W_{LIN}}||^2 = \frac{1}{N}||(I-XX^{\dagger})y||^2$$

其中I是$N\times N$的单位矩阵，$XX^{\dagger}$ 可以使用$N\times N$的H矩阵（hat matrix）表示。

此处通过几何图形来更具体的了解H矩阵的物理意义，如果9-4所示。

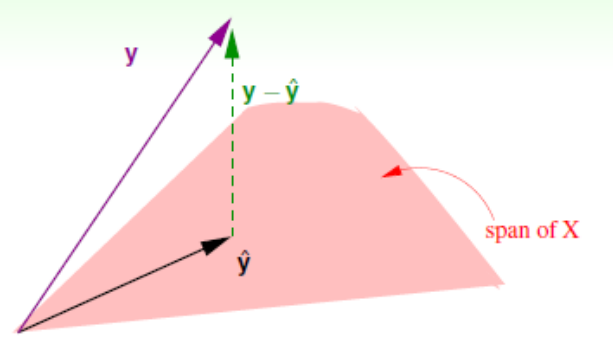


图9-4 有关H矩阵的几何图形

其中紫色向量表示实际输出向量y。

粉色区域表示输入矩阵X乘以不同权值向量w所构成的空间，从这个定义得知，解析解求得最优权值向量$W_{LIN}$所表示的输出向量$\hat{y} = Xw_{LIN}$ 也落在该空间中，其中$\hat{y}$ 也N维向量，不难想象$\hat{y}$正是实际输出向量y在该空间上的投影。

而绿色虚线表示实际输出与最优假设输出之间的差距，写作$y - \hat{y}$。 从上述情况可知，$y - \hat{y} \perp$粉色空间

因此得知H矩阵是一个投影过程，向量$\hat{y}$ 是向量y通过矩阵H所做的投影，可以将矩阵H理解为一系列旋转缩放的动作，有$\hat{y} = Hy$。$I - H$ 的矩阵同样是一个投影过程，$y - \hat{y}$ 是向量y通过$I - H$ 的线性变化得到的向量。

在图9-4中再加入一点元素，构成图9-5。

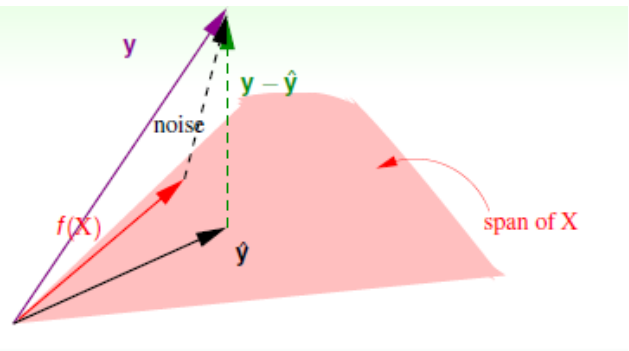


图9-5 加入理想的目标输出f(x)

如果实际输出矩阵y由理想的目标输出f(x)加上噪音部分共同构成（如图中红色和黑色虚线部分）,则其中$y - \hat{y}$的形式也可以通过噪音按照$I - H$的变换方式构成。因此，得到公式9-16所示。

$$E_{in}(W_{LIN}) = \frac{1}{N}||y - \hat{y}||^2 = \frac{1}{N}||(I - H)noise||^2 = \frac{1}{N}(N - (d+1))||noise||^2$$

其中$N - (d+1)$ 是通过$I - H$的迹（trace）得出的。在求解之前，可以想象$(I - H)^2 = I - H$，因为经过两次转换所得到的还是第一次时的误差向量，trace(I-H)的求解过程如公式9-17所示。

$$trace(I - H) = trace(I) - trace(H)$$

(根据迹的性质，$trace(A + B) = trace(A) + trace(B)$)

$$= N - trace(XX^{\dagger}) = N - trace(X(X^TX)^{-1}X^T) = N - trace(X^TX(X^TX)^{-1})$$

(根据迹的性质$trace(ABC) = trace(CAB)$)

$$= N - trace(\underbrace{I}_{(d+1)\times(d+1)}) = N - (d+1) \quad \text{公式9-17}$$

最终介绍下该I-H这种转换的物理意义：原来有一个有N个自由度的向量y，投影到一个有d+1维的空间x（代表一列的自由度，即单一输入样本的参数），而余数剩余的自由度最大只有N-(d+1)种。

最终可以写出$\vec{E}_{in}$的结果，同理也可以写出 $\vec{E}_{out}$的噪音表示，如公式9-18和公式9-19所示。

$\vec{E}_{in} = noiselevel \cdot (1 - \frac{d+1}{N})$ 公式9-18

$\vec{E}_{out} = noiselevel \cdot (1 + \frac{d+1}{N})$ 公式9-19

因为前者做了优化，所以 $\vec{E}_{in}$ 有机会比「理想值」多 fit 数据中的 噪音一点点，所以会比「理想值」好一点；但 $\vec{E}_{out}$的部份则要付出代价(想象在测试的时候，拿到了与训练数据「完全不同」的噪音)，所以可能反而多远离理想值一些。

从上面两个公式可以得到一个学习鸿沟图，如图9-6所示。

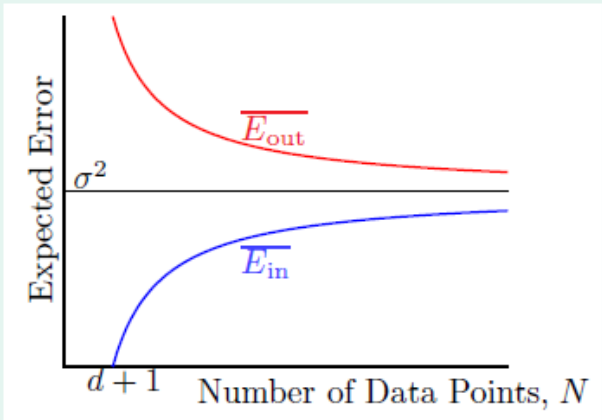


图9-6 机器学习的学习鸿沟

其中N在趋于无穷大时，$\vec{E}_{in}$ 与$\vec{E}_{out}$两者都会趋近于noise level的值，即$\sigma^2$

泛化误差之间的差距：$\frac{2(d+1)}{N}$

至此可以表明在线性回归中可以寻找到很好的$E_{out}$，因此线性回归可以学习。

## 9.4 Linear Regression for Binary Classification 使用线性回归做二元分类

首先对比二元线性分类与线性回归之间的差异，分别在三个部分进行对比，输出空间、假设函数和错误衡量函数，如图9-7所示。

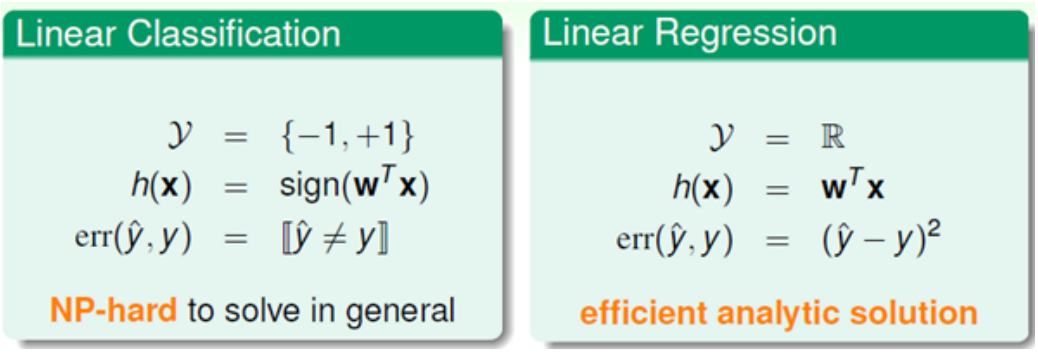


图9-7 二元线性分类与线性回归的对比

从求解问题的难度考虑，二元分类的求解是一个NP难问题，只能使用近似求解的方式，而线性回归通过求解析解，求解方便，程序编写也简单。

因此考虑能否通过求解线性回归的方式求二元分类问题，因为二元分类的输出空间{-1，+1}属于线性回归的输出空间，即$\{-1, 1\} \in R$。其中数据集的标记大于零的表示+1，小于零的表示-1，通过线性回归求得的解析解$E_{LIN}$,直接得出最优假设 $g(x) = sign(W_{LINX})$。但是这种推理只符合直觉，而如何使用数学知识去说明这种方式的合理性呢？

观察两种错误衡量方式，$err_{0/1}$和分$err_{lr}$别表示为公式9-20和公式9-21。

$err_{0/1} = [|sign(w^T x) \neq y|]$ 公式9-20

$err_{lr} = (w^T x - y)^2$ 公式9-21

观察两公式的共同特点都含有$w^T x$这一向量内积的形式，如果将作$w^T x$为横轴，将err结果作为纵轴，可以画出图9-8。

其中图9-8a）为y=+1时，两err值的图像表示；而图9-8b）为y=-1时，两err值的图像表示。两幅图中红色的线表示$err_{lr}$，蓝色的先表示$err_{0/1}$。

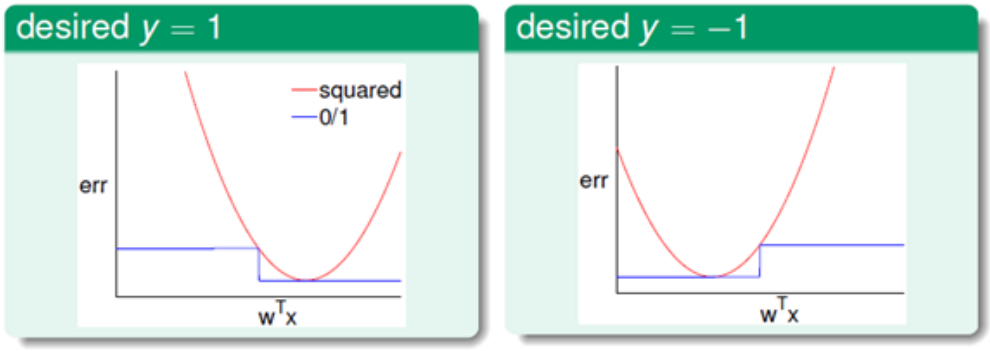


图9-8 a)y=+1时，两err值 的表示 b) y=-1时两err值的表示

从图中得出公式9-22的结论。

$$err_{0/1} \leq err_{lr}$$

回忆下第七章中证明的二元分类下$E_{out}$的上限，结合公式9-22的结论，得公式9-23。

$$classification E_{out}(W) \leq classification E_{in}(W) + \sqrt{\frac{8}{N}\ln(\frac{4(2N)^{d_{VC}}}{\delta})} \leq regression E_{in}(W) + \sqrt{\frac{8}{N}\ln(\frac{4(2N)^{d_{VC}}}{\delta})}$$

因此二元分类问题得到了一个更宽松的上界，但是也是一种更有效率的求解方式。

在实际运用中，一般都将通过线性回归求得的解析解$W_{LIN}$作为PLA或者pocket的初始值$W_0$达到快速求解的目的 。