基本形式

线性模型通常表示如下：

$$f(x) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$$

向量形式如下：

$$f(x) = w^Tx + b$$

线性模型形式简单，却蕴含着机器学习中的重要思想（w直观表达了各属性在预测中的重要性），许

多强大的非线性模型就是在线性模型的基础上通过引入层次结构或高纬映射而得

线性回归

通常给定数据集$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$,其中$x_i = (x_{i1}; x_{i2}; \ldots; x_{id})\$, y_i \in R$

线性回归试图学得

$$f(x_i) = wx_i + b \quad s.t.\ f(x_i) \approx y_i$$

一维情形 (d=1)

模型的学习只需确定参数w和b，使用最小二乘估计有目标函数：

$$（w^*, b^*）= arg\,min_{(w,b)} \sum_{i=1}^{n} (y_i - wx_i - b)^2$$

通过求偏导可以得到估计

$$w^* = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$b^* = \bar{y} - w\bar{x}$$

多维情形 (d>1)

为了便于分析，我们将w, b吸收入向量形式$\hat{w} = (w; b)$，把数据集D表示为一个$n \times (d+1)$的矩阵X

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} & 1 \end{bmatrix}=\begin{bmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_n^T & 1 \end{bmatrix}$$

把标记也写成向量形式如下：

$$y=\begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T$$

于是类似地我们可得到目标函数：

$$\hat{w}^*=arg\,min_w(y-X\hat{w})^T(y-X\hat{w})$$

求导计算得到估计

$$\hat{w}^*=(X^TX)^{-1}X^Ty$$

广义线性回归

考虑单调可微函数$g(\cdot)$，令

$$y=g^{-1}(w^Tx+b)$$

这样得到的模型称为“广义线性模型”，其中对数线性回归是广义线性模型在$g(\cdot)=ln(\cdot)$时的特例

逻辑斯蒂回归（logistics regression）

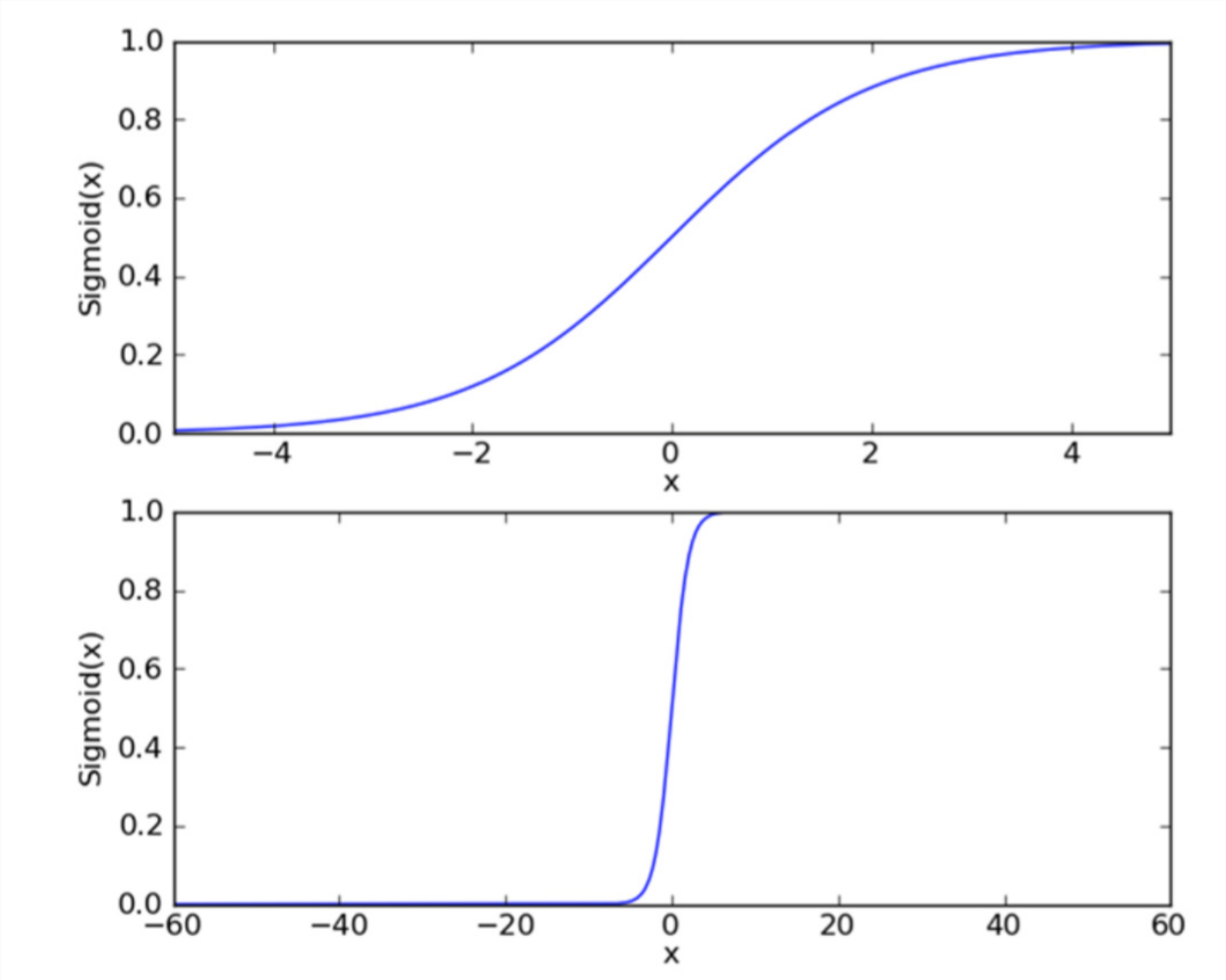
由线性回归到分类，考虑二分类任务，记其输出标记$y\in\{0,1\}$，而线性回归模型产生的预测值

$z=w^Tx+b$是实值，于是我们需要将实值 z 转化为0/1值，例如利用函数

$$y=\begin{cases} 0, & z<0 \\ 0.5, & z=0 \\ 1, & z>0 \end{cases}$$

在逻辑斯蒂回归中我们使用了Sigmoid函数

$$y=\frac{1}{1+e^{-z}}$$



于是我们得到

$$y = \frac{1}{1 + e^{-(w^T x + b)}}$$

变形可得

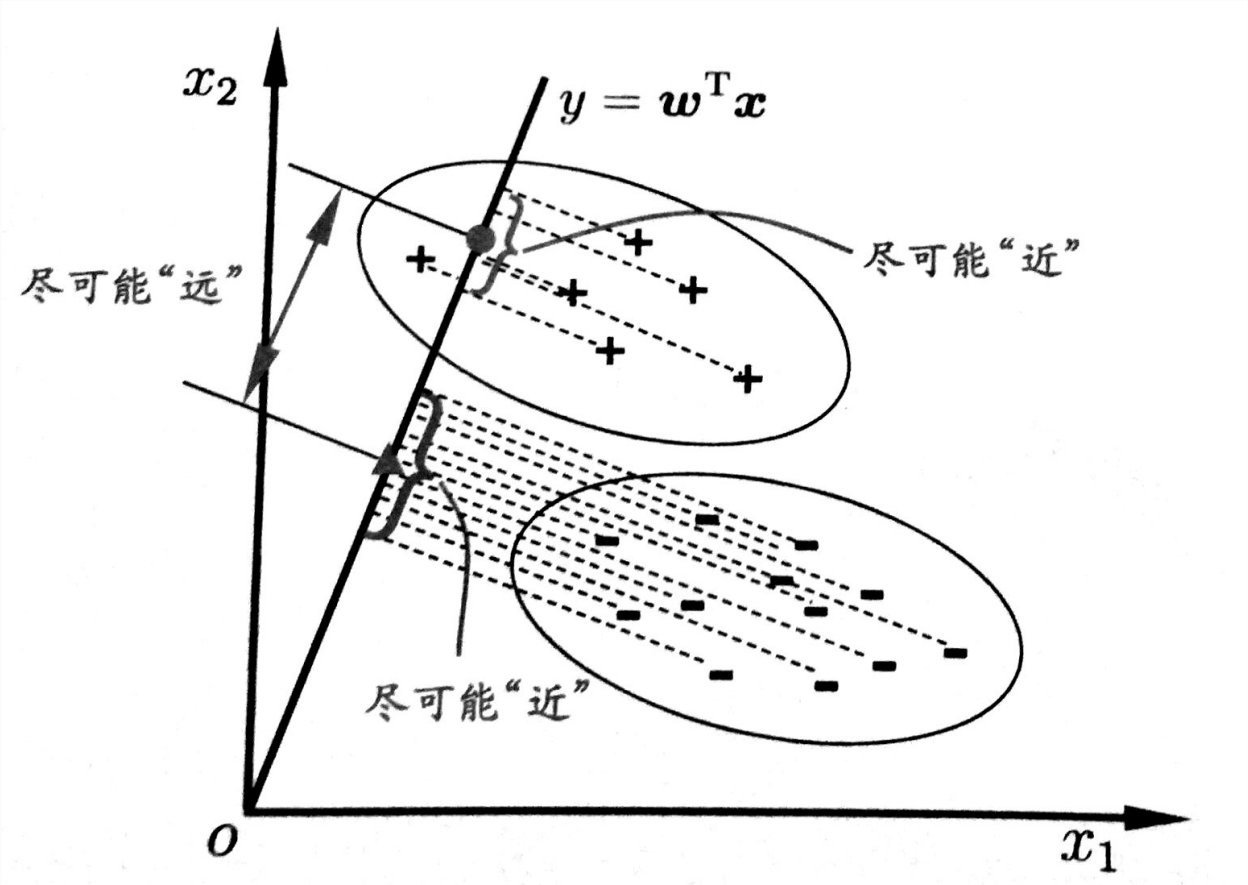
$$ln(\frac{y}{1-y}) = w^T x + b$$

若将y视作将样本x作为正例的可能性，那么$1 - y$是其反例可能性，两者的比值称为几率，取对数后

则得到对数几率，所以，上式是在用线性回归模型的预测结果去逼近真实标记的对数几率

线性判别分析（LDA）

LDA(Linear Discriminant Analysis)的思想：给定训练样例集，设法将样例投影到一条直线上，使得同类样例的投影点尽可能近、异类样例的投影点尽可能远；在对新样例进行分类时，将其投影到同样的这条直线上，再根据投影点的位置来确定新样本的类型，图形示例如下：



欲使同类样例的投影点尽可能接近，可以让同类样例投影点的协方差尽可能小，而欲使异类样例的投影点尽可能远，可以让类中心之间的距离尽可能大，依据这两点可以构建优化目标函数并求解

多分类学习

基本思路：将多分类任务拆解成若干个二分类任务求解

拆分策略：One vs. One / One vs. Rest / Many vs. Many

OvO

给定数据集$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, y_i \in \{C_1, C_2, \ldots, C_N\}$.OvO将这N个类别两两配对，从而产生N(N-1)/2个二分类任务。例如OvO将为区分类别C_i, C_j训练一个分类器，该分类器把D中的C_i类样例作为正例，C_j类样例作为反例。在测试阶段，新样本将同时提交给所有的分类器，于是我们将得到$N(N-1)/2$个分类结果，最终把预测得最多的类别作为最终分类结果。

OvR

OvR则是每次将一个类的样例作为正例，其余的所有类的样例作为反例来训练N个分类器，其余同OvO.