4 Moving Beyond Linearity 非线性回归

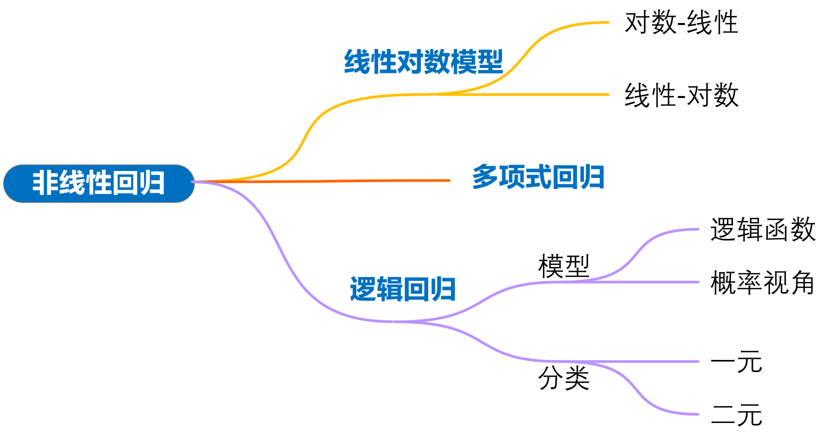
寻找因变量和自变量之间关系的非线性模型

科学不去尝试辩解，甚至几乎从来不解读；科学主要工作就是数学建模。模型是一种数学构造；基于少量语言说明，每个数学构造描述观察到的现象。数学模型合理之处是它具有一定的普适性；此外，数学模型一般具有优美的形式——也就是不管它能解释多少现象，它必须相当简洁。

The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.

—— 约翰·冯·诺伊曼 (John von Neumann) | 美国籍数学家 | 1903 ~ 1957

- matplotlib.pyplot.contour() 绘制等高线线图
- matplotlib.pyplot.contourf() 绘制填充等高线图
- matplotlib.pyplot.getp() 获绘图对象的属性
- matplotlib.pyplot.plot_wireframe() 绘制线框图
- matplotlib.pyplot.scatter() 绘制散点图
- matplotlib.pyplot.setp() 设置绘图对象的一个或者多个属性
- numpy.random.normal() 产生服从高斯分布的随机数
- numpy.random.rand() 产生服从均匀分布的随机数
- numpy.random.randn() 产生服从标准正态分布的随机数
- scipy.special.expit()
- seaborn.jointplot() 绘制联合分布/散点图和边际分布
- seaborn.kdeplot() 绘制概率密度估计曲线
- seaborn.scatterplot() 绘制散点图
- sklearn.linear_model.LinearRegression() 最小二乘法回归
- sklearn.linear_model.LogisticRegression() 逻辑回归函数，也可以用来分类
- sklearn.pipeline.Pipeline() 将许多算法模型串联起来形成一个典型的机器学习问题工作流
- sklearn.preprocessing.FunctionTransformer() 根据函数对象或者自定义函数处理样本数据
- sklearn.preprocessing.PolynomialFeatures() 建模过程中构造多项式特征


非线性回归
线性对数模型
对数-线性
线性-对数
多项式回归
逻辑回归
模型
逻辑函数
概率视角
分类
一元
二元

4.1 线性回归

本书前文介绍过线性回归，白话说，线性回归使用直线、平面或超平面来预测。多元线性回归的数学表达式如下：

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_D x_D + \varepsilon \tag{1}$$

可以发现 $x_1, x_2, \ldots, x_D$ 这几个变量的次数都是一次，这也就是“线性”一词的来由。图 1 所示为最小二乘法多元线性回归数据关系。

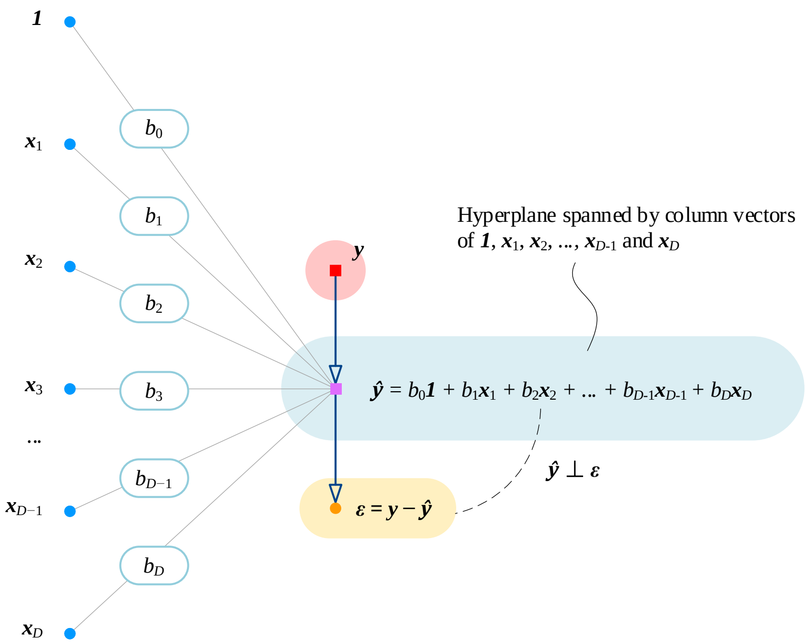


图 1. 最小二乘法多元线性回归数据关系

此外，特征还可以进行线性组合得到一系列新特征：

$$\boldsymbol{z}_k = v_{1,k}\boldsymbol{x}_1 + v_{2,k}\boldsymbol{x}_2 + \cdots + v_{D,k}\boldsymbol{x}_D = \phi_k\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_D\right) \tag{2}$$

即

$$\begin{aligned} \boldsymbol{Z} &= \begin{bmatrix} \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_p \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{X}) & \cdots & \phi_p(\boldsymbol{X}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D \end{bmatrix} \begin{bmatrix} v_{1,1} & \cdots & v_{1,p} \\ v_{2,1} & \cdots & v_{2,p} \\ \vdots & \ddots & \vdots \\ v_{D,1} & \cdots & v_{D,p} \end{bmatrix} \end{aligned} \tag{3}$$

然后可以用最小二乘求解回归系数：

$$\hat{\boldsymbol{y}} = \boldsymbol{Z}\left(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{y} \tag{4}$$

图 2 所示为线性组合的数据关系，得到的模型可以通过 (3) 反推得到基于 $x_1, x_2, \ldots, x_D$ 这几个变量的线性模型。本书后续介绍的基于主成分分析的回归方法采用的就是这一思路。

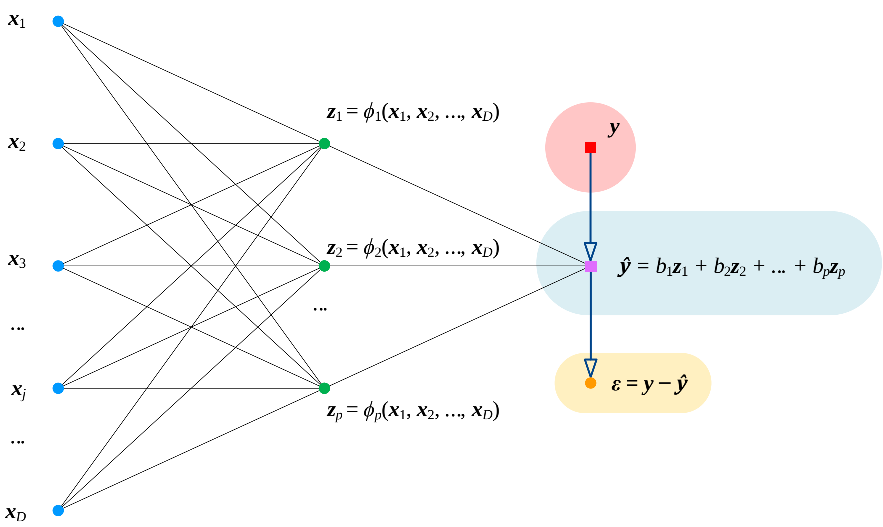


图 2. 特征线性组合

线性回归虽然简单，但是并非万能。图 3 给出的三组数据都不适合用线性回归来描述。本章就介绍如何采用几种非线性回归方法来解决线性回归不能解决的问题。

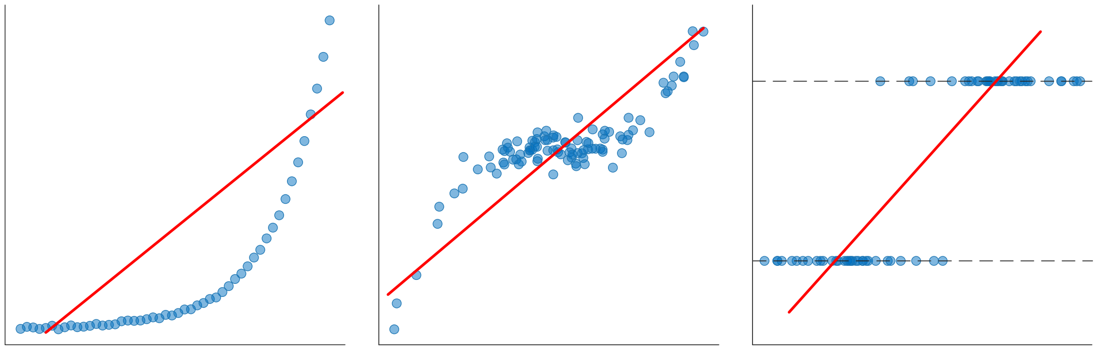


图 3. 线性回归失效的三个例子

4.2 线性对数模型

本书前文介绍过数据转换，一些回归问题可以对输入或输出进行数据转换，甚至对两者同时进行数据转换，之后再来构造线性模型。本节介绍几个例子。

观察图 4 (a)，容易发下样本数据呈现出“指数”形状，而且输出值 y 大于 0；容易想到对输出值 y 取对数，得到图 4 (b)。而图 4 (b) 展现出明显的线性回归特征，便于进行线性回归建模。

利用以上思路便可以得到所谓对数-线性模型：

$$\ln y = b_0 + b_1 x + \varepsilon \qquad (5)$$

图 5 所示为通过拟合得到的对数-线性模型。

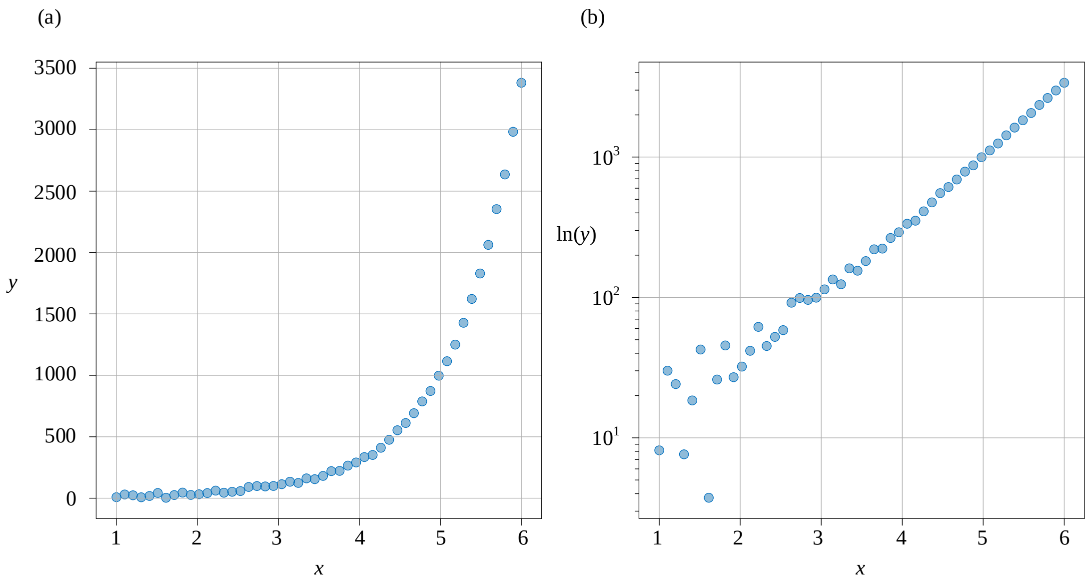


图 4. 类似“指数”形状的样本数据

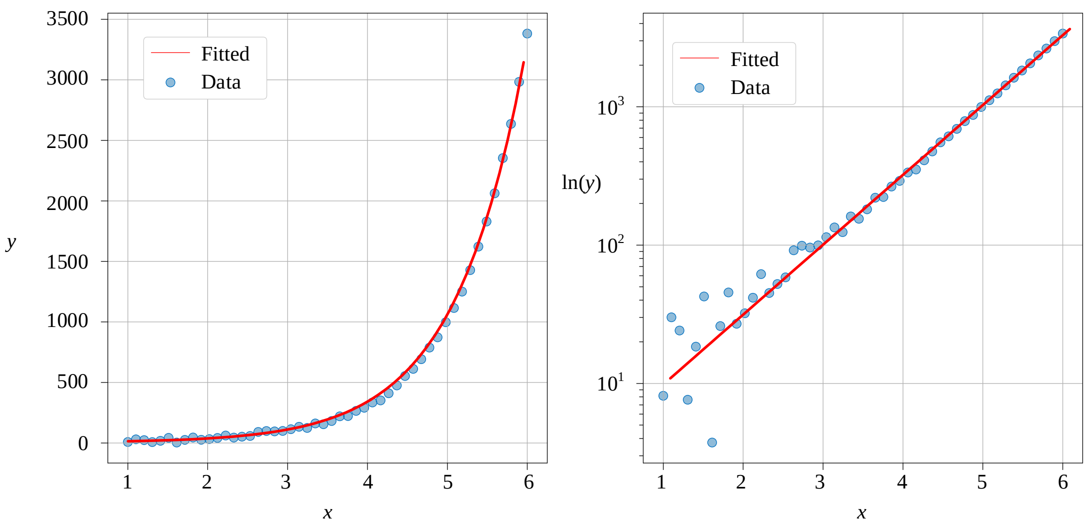


图 5. 对数-线性模型

反过来，当数据呈现类似“对数”形状时 (见图 6 (a))，可以对输入 x 去对数，得到图 6 (b)。观察图 6 (b)，可以发现数据展现出一定的线性关系。这样我们就可以使用线性-对数模型：

$$y = b_0 + b_1 \ln x + \varepsilon \tag{6}$$

图 7 所示为得到的线性-对数模型。

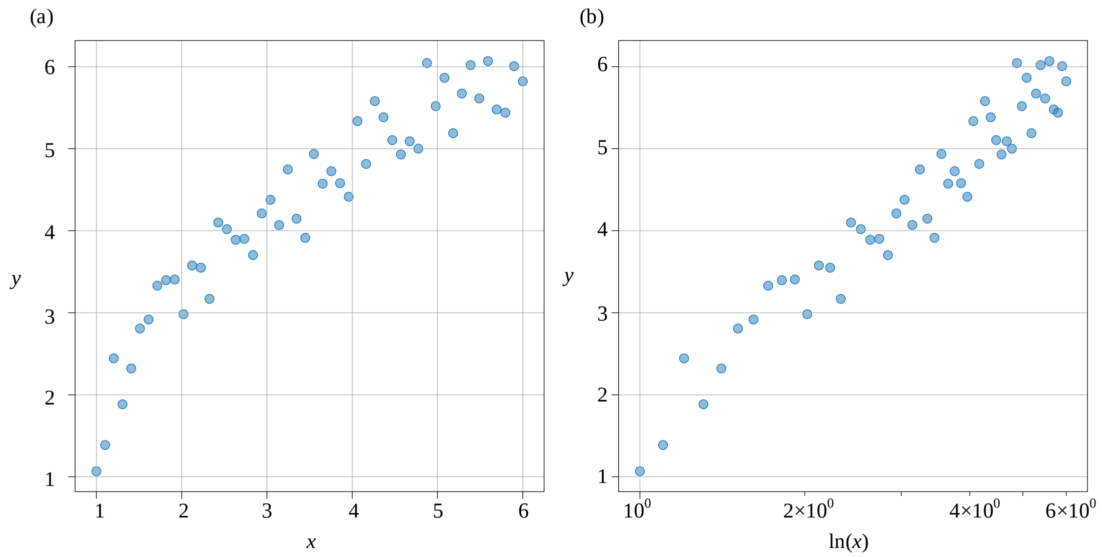


图 6. 类似“对数”形状的样本数据

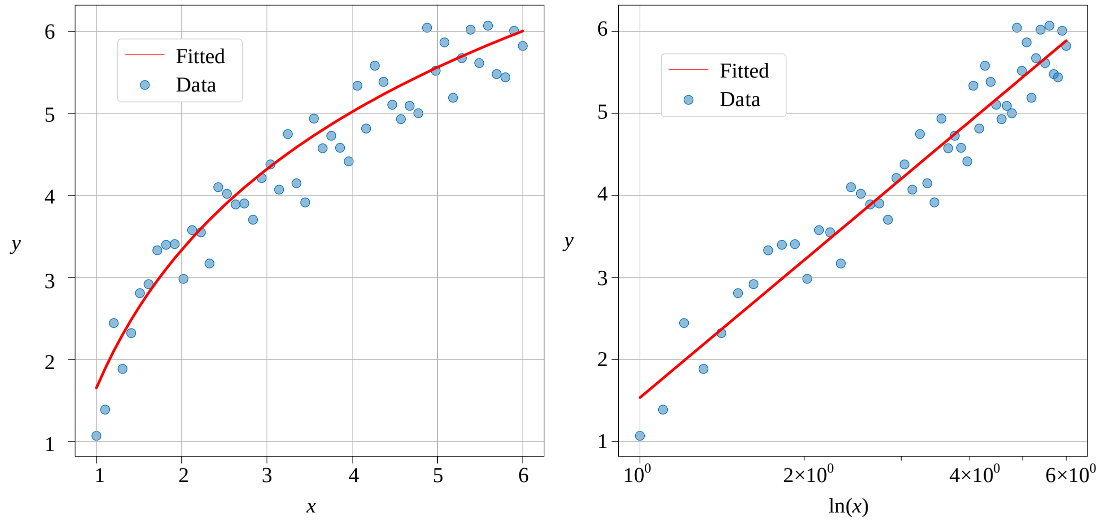


图 7. 线性-对数模型

此外，我们可以理解同时对输入和输出数据取对数，然后再构造线性回归模型；这种模型叫做双对数模型：

$$\ln y = b_0 + b_1 \ln x + \varepsilon \tag{7}$$

需要注意的是，进行对数变换的前提是，所有的观测值都必须大于 0。当观测值中存在 0 或者小于 0 的数值，可以对所有的观测值加 $-\min(x) + 1$，然后再进行对数变换。

Bk6_Ch13_01.py 绘制本节图像。

4.3 非线性回归

非线性回归是一种回归分析方法，建立自变量与因变量之间的非线性关系模型，用于预测连续变量的值。非线性回归需要应对线性回归无法解决的复杂问题。

有些情况下，简单的将数据做对数处理是不够的，需要对数据做进一步处理。模型如下所示：

$$y = f(x) + \varepsilon \tag{8}$$

$f(x)$ 可以是任意函数，比如多项式函数，逻辑函数，甚至是分段函数。

(8) 中 $f(x)$ 可以是多项式，得到**多项式回归** (polynomial regression)。比如，一元三次多项式回归：

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \tag{9}$$

图 8 所示为一元三次多项式回归模型数据关系。

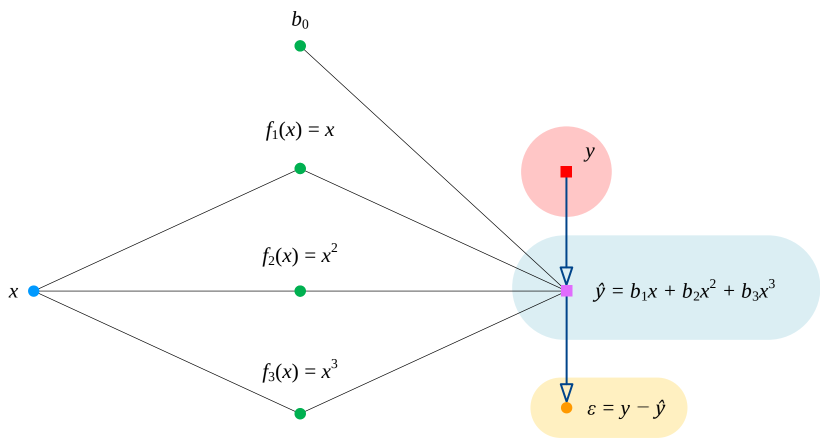


图 8. 一元三次多项式回归

图 9 所示为利用一元三次多项式回归模型来拟合样本数据。下一节，我们将仔细讲解多项式回归。

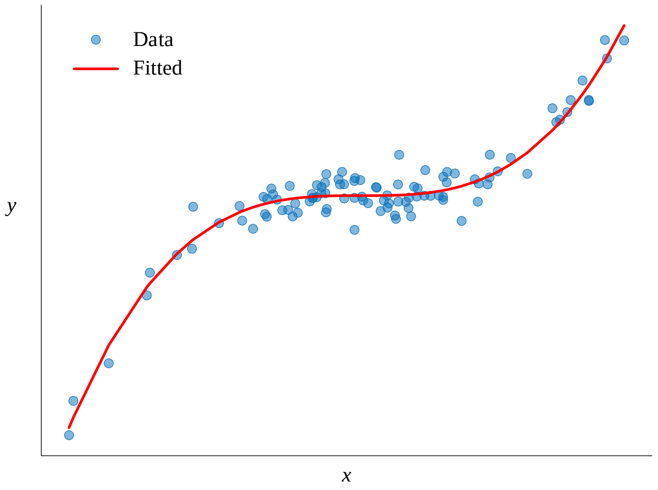


图 9. 一元三次多项式回归模型

逻辑回归 (logistic regression) 也是一种重要的非线性回归模型。一元逻辑回归模型如下：

$$y=\frac{1}{1+\exp\left(-\left(\underbrace{b_0+b_1x}_{\text{linear model}}\right)\right)} \tag{10}$$

图 10 所示为拟合数据得到的逻辑回归模型。图 11 所示为逻辑回归模型数据关系，逻辑回归模型可以看做时线性模型通过逻辑函数转换得到。

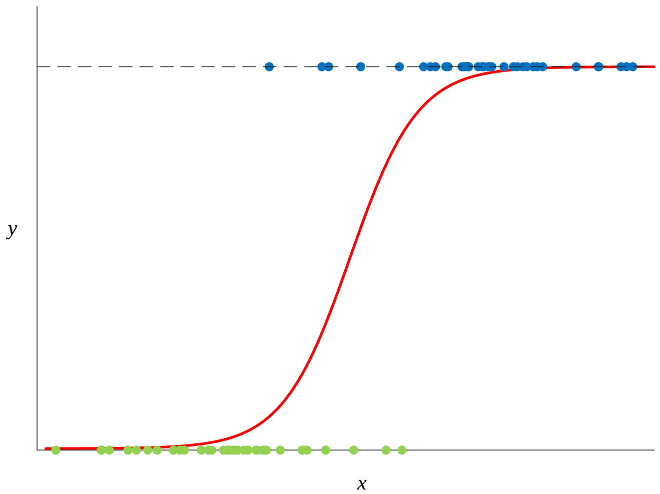


图 10. 逻辑回归模型

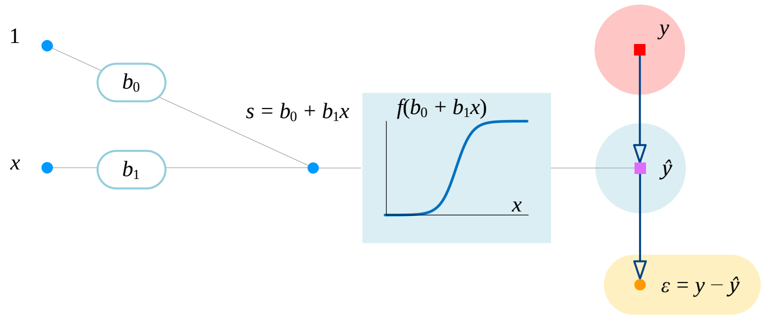


图 11. 逻辑回归数据关系

逻辑回归虽然是个回归模型，但是常被用作分类模型，用于二分类。

下一章将讲解逻辑回归。

此外，我们还可以用分段函数来拟合数据。如图 12 所示，两段线性函数用来拟合样本数据，效果也是不错的。

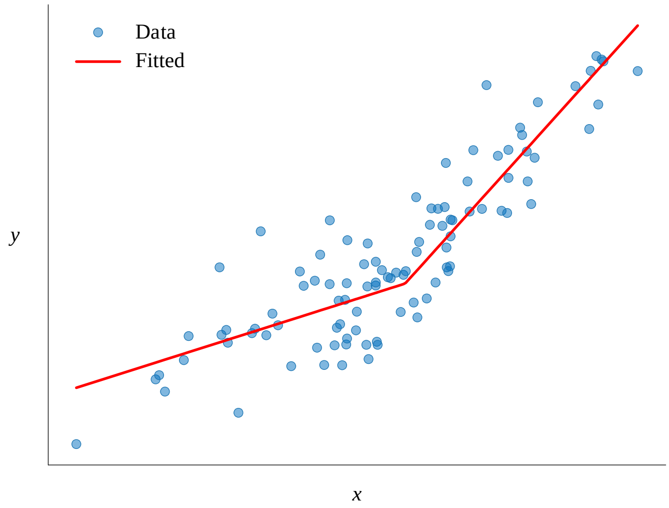


图 12. 分段函数模型

非参数回归 (non-parametric regression) 也是一种非常重要的非线性拟合方法。本章前面介绍的回归模型都有自身的"参数"，但是非参数回归模型并不假设回归函数的具体形式。参数回归分析时假定变量之间某种关系，然后估计参数；而非参数回归，则让数据本身说话。

比如，图 13 所示为采用**最邻近回归** (k-nearest neighbor regression)。最邻近可以用来分类，也可以用来构造回归模型。

鸢尾花书《机器学习》一书讲介绍最邻近方法。

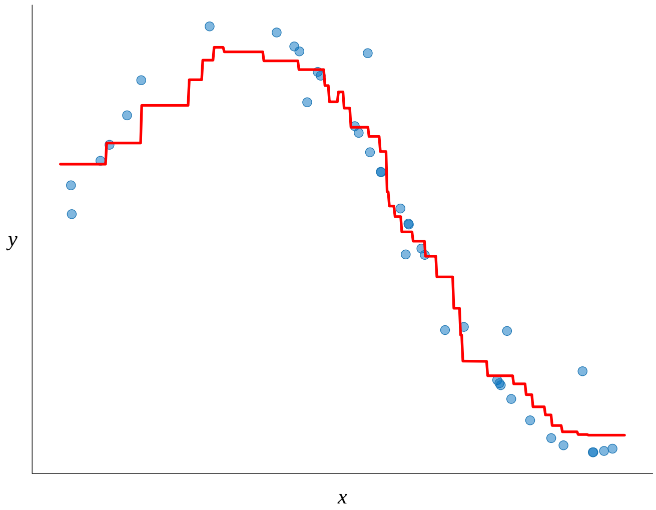


图 13. 最邻近回归

4.4 多项式回归

多项式回归是回归分析的一种形式，多项式回归是指回归函数的自变量的指数大于 1。在多项式回归中，一元回归模型最佳拟合线不是直线，而是一条拟合了数据点的多项式曲线。

图 14 所示为第一到五次一元函数的形状。

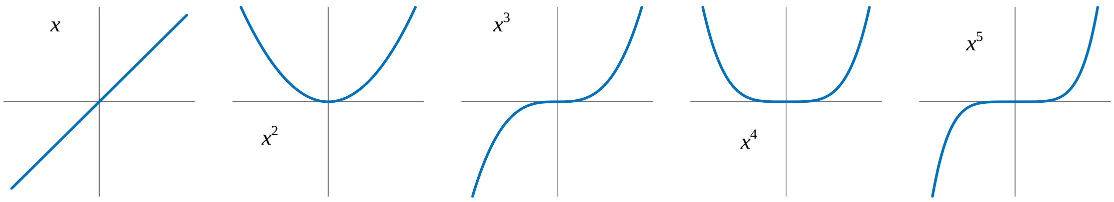


图 14. 一次到五次一元函数

自变量 x 和因变量 y 之间的关系被建模为关于 x 的 m 次多项式：

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \cdots + b_m x^m \tag{11}$$

其中，m 为多项式函数最高次项系数。

图 15 所示为一元多项式回归数据关系。

《矩阵力量》第 9 章介绍过采用矩阵运算得到多项式回归系数，请大家回顾。

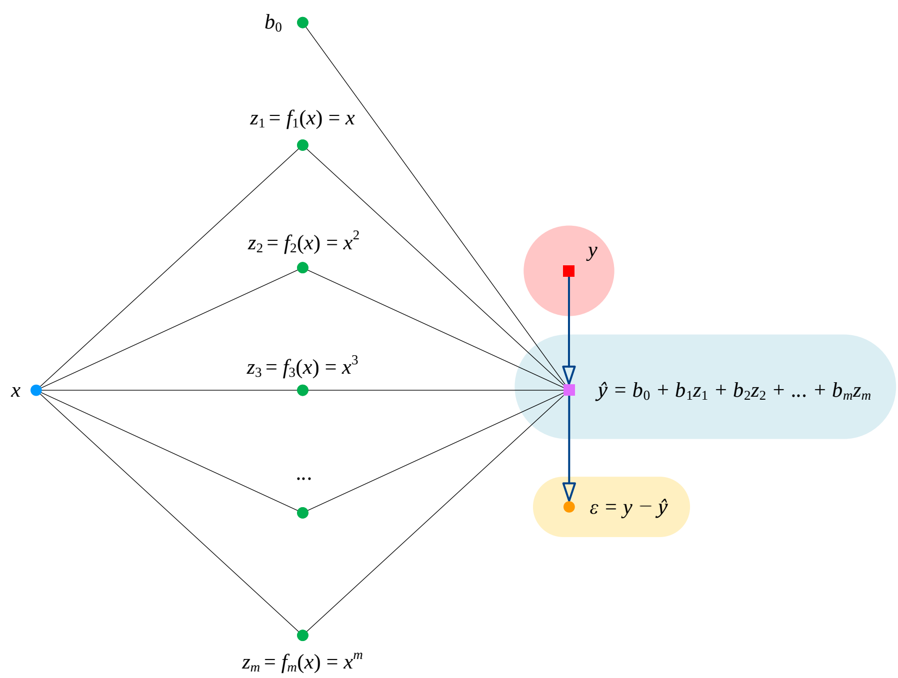


图 15. 一元多项式回归数据关系

图 16 所示为采用一次到四次一元多项式回归模型拟合样本数据。多项式回归的最大优点就是可以通过增加自变量的高次项对数据进行逼近。

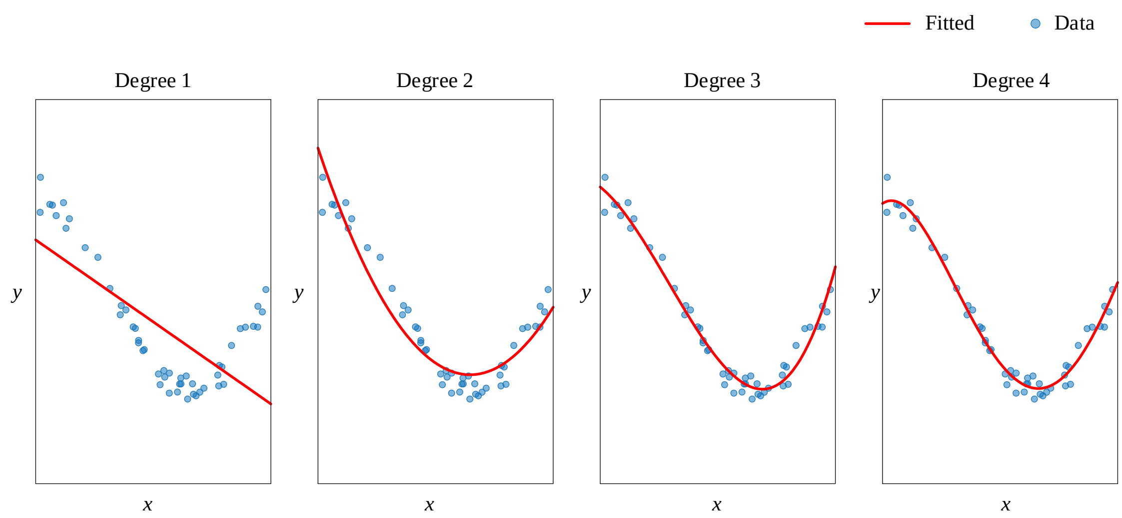


图 16. 一元多项式回归，一次到四次

但是，对于多项式回归，次数越高，越容易产生过度**拟合** (overfitting) 问题。过拟合发生的原因是，使用过于复杂的模型，导致模型过于精确地描述训练数据。如图 17 所示，采用过高次数的多项式回归模型，模型过于复杂，过度捕捉训练数据中的细节信息，甚至是噪音。但是，使用该模型预测其他样本数据时，会无法良好地预测未来观察结果。丛书后续还要深入探讨过拟合问题。

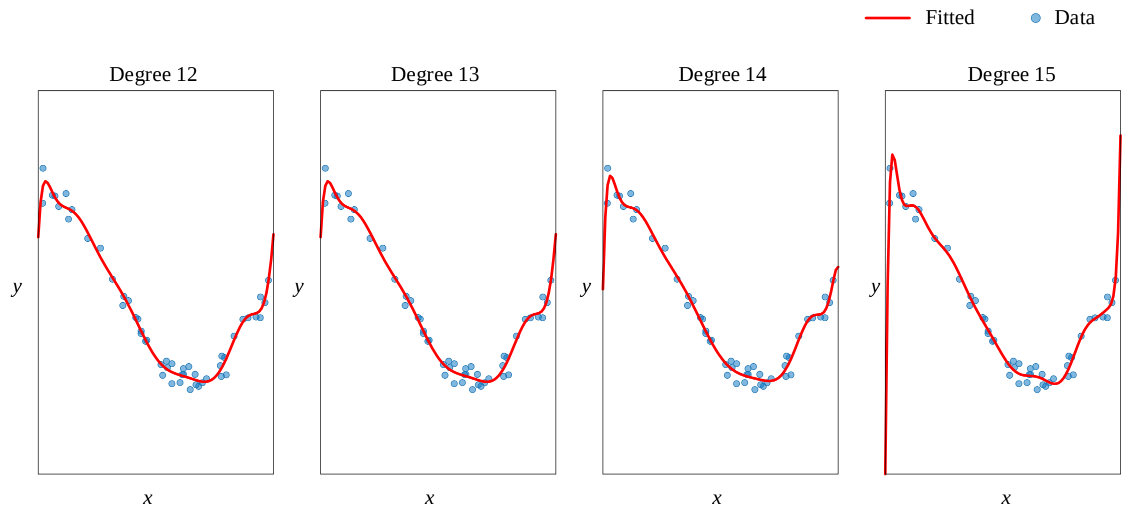


图 17. 一元多项式回归过度拟合，12 次到 15 次

此外，多项式回归可以有多个特征，而特征和特征之间可以形成较为复杂的多项式关系。比如，下式给出的是二元二次多项式回归：

$$f(x_1, x_2) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2 + b_4 x_1^2 + b_5 x_2^2 \tag{12}$$

(12) 相当于以一定比例组合图 18 所示的六个平面。提高多项式项次数，可以获得更加复杂的曲线或曲面，这样可以描述更加复杂的数据关系。因此不论因变量与其它自变量的关系如何，一般都可以尝试用多项式回归来进行分析。

图 19 所示为 (12) 所示的数据关系。

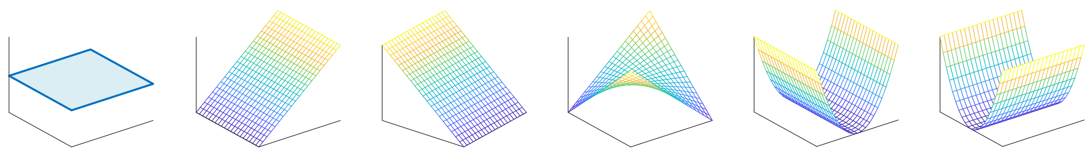


图 18. 六个二元平面/曲面

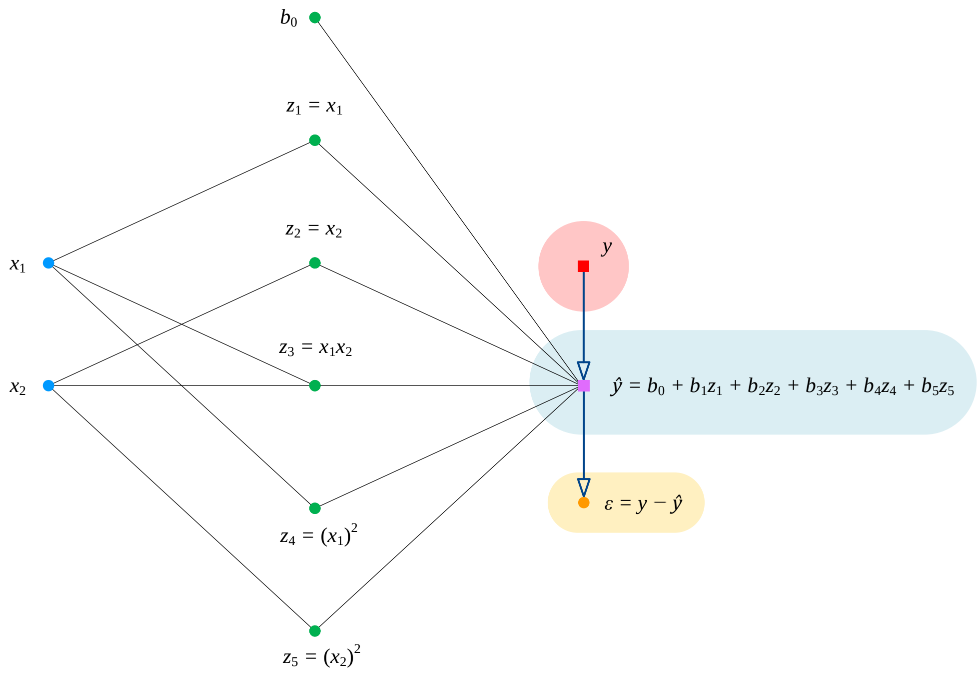


图 19. 二元二次多项式回归数据关系

Bk6_Ch13_02.py 绘制本节图像。

4.5 逻辑回归

图 20 给出一组数据的散点图，取值为 1 的数据点被标记为蓝色，取值为 0 的数据点被标记为红色。图 21 给出三种可以描述红蓝散点数据的函数。线性函数显然不适合这一问题。阶跃函数虽然可以捕捉函数从 0 到 1 的跳变，但是函数本身不光滑。

逻辑函数似乎能够胜任描述红蓝三点数据的任务。线性函数的因变量一般为连读数据；而逻辑函数的因变量为离散数值，即分类数据。

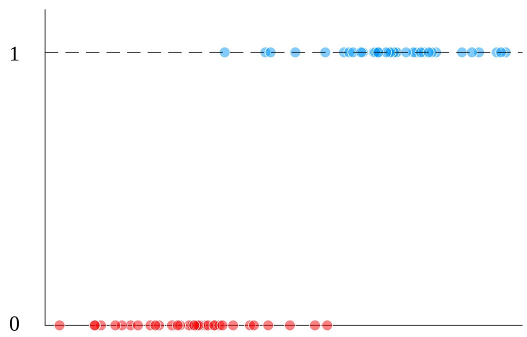


图 20. 红蓝数据的散点图

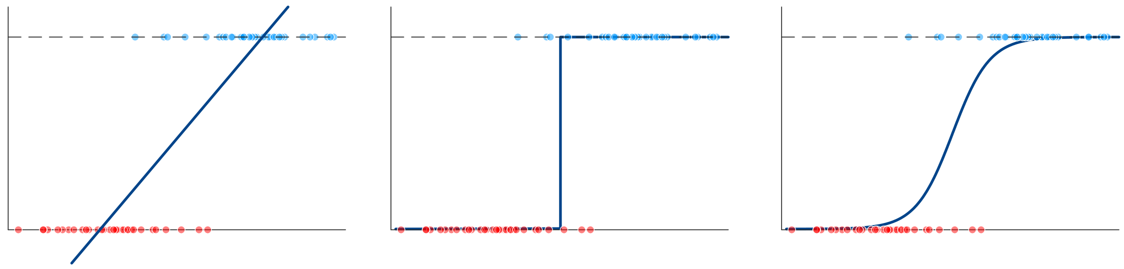


图 21. 可以描述红蓝数据的函数

逻辑函数

回顾《数学要素》12 章讲过的逻辑函数。

最简单的逻辑函数：

$$f(x)=\frac{1}{1+e^{-x}}=\frac{e^{x}}{1+e^{x}} \tag{13}$$

更一般的一元逻辑函数：

$$f(x)=\frac{1}{1+\exp\left(-\left(b_0+b_1x\right)\right)} \tag{14}$$

图 22 所示为 b_1 影响一元逻辑函数图像的陡峭程度。图中，$b_0 = 0$。可以发现函数呈现 S 形，取值范围在 [0, 1] 之间；函数在左右两端无限接近 0 或 1。函数的这一性质，方便从概率角度解释，这是下一节要介绍的内容。

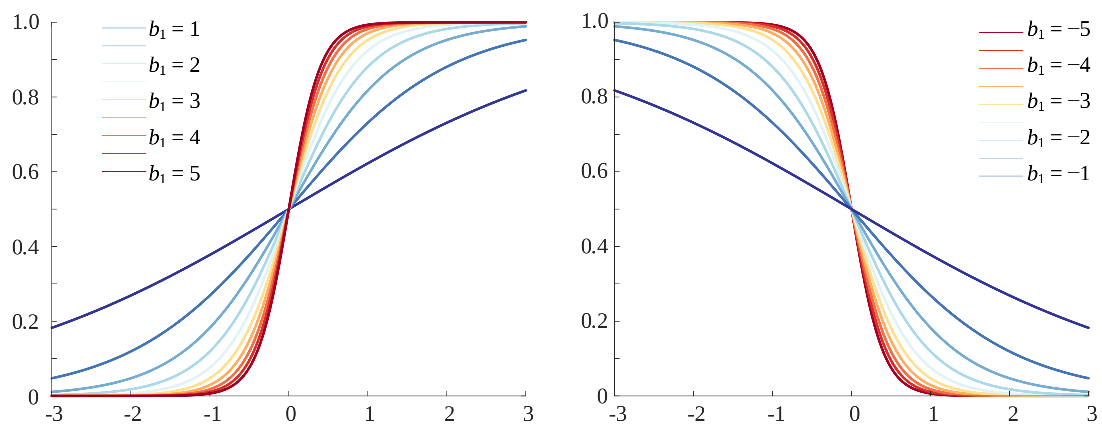


图 22. b_1 影响一元逻辑函数图像的陡峭程度

找到 $f(x) = 1/2$ 位置：

$$f(x)=\frac{1}{1+\exp\left(-\left(b_0+b_1x\right)\right)}=\frac{1}{2} \tag{15}$$

整理得到 $f(x) = 1/2$ 对应的 x 值：

$$x = -\frac{b_0}{b_1} \tag{16}$$

也就是当 b_1 确定时，b_0 决定逻辑函数位置。注意，图 23 中，$b_1 = 0$。

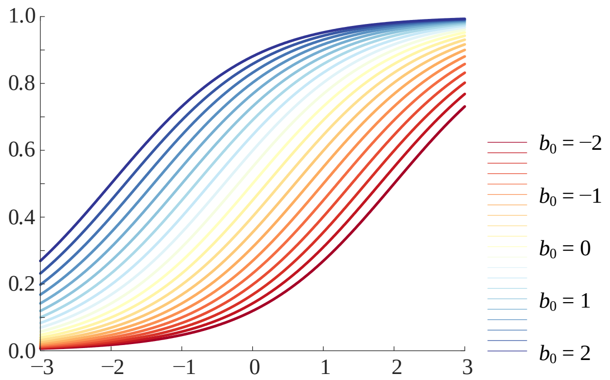


图 23. b_0 决定逻辑函数位置，$b_1 = 0$

图 24 所示为根据数据的分布，选取不同的逻辑函数参数。

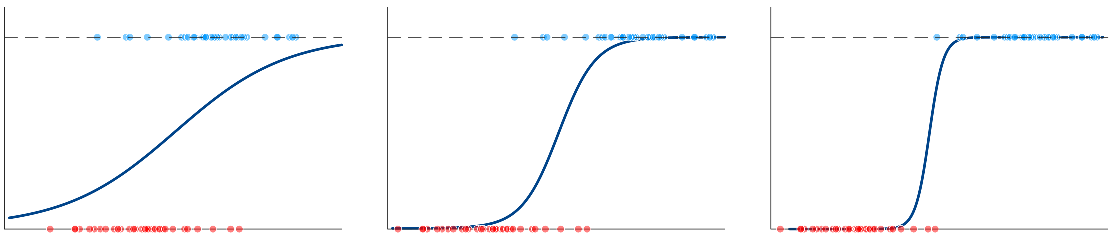


图 24. 根据数据的分布，选取不同的逻辑函数参数

Bk6_Ch13_03.py 绘制逻辑函数图像。

多元

对于多元情况，逻辑函数的一般式如下：

$$f(x_1, x_2, \ldots, x_D) = \frac{1}{1 + \exp\left(-\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots b_D x_D\right)\right)} \tag{17}$$

利用矩阵运算表达多元逻辑函数：

$$f(\boldsymbol{x})=\frac{1}{1+\exp\left(-\boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}\right)} \tag{18}$$

其中

$$\begin{aligned}\boldsymbol{x}&=\begin{bmatrix}1 & x_1 & x_2 & \cdots & x_D\end{bmatrix}^{\mathrm{T}}\\ \boldsymbol{b}&=\begin{bmatrix}b_0 & b_1 & b_2 & \cdots & b_D\end{bmatrix}^{\mathrm{T}}\end{aligned} \tag{19}$$

令

$$s(\boldsymbol{x})=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}=b_0+b_1x_1+b_2x_2+\cdots b_Dx_D \tag{20}$$

(18) 可以记做：

$$f(s)=\frac{1}{1+\exp(-s)} \tag{21}$$

(20) 相当于是线性回归，经过如 (21) 逻辑函数映射，得到逻辑回归。图 25 所示为逻辑回归和线性回归之间关系。图 25 这幅图已经让我们看到**神经网络** (neural network) 的一点影子，逻辑函数 $f(s)$ 类似**激活函数** (activation function)。

特别地，对于二元逻辑函数：

$$f(x_1,x_2)=\frac{1}{1+\exp\left(-\left(b_0+b_1x_1+b_2x_2\right)\right)} \tag{22}$$

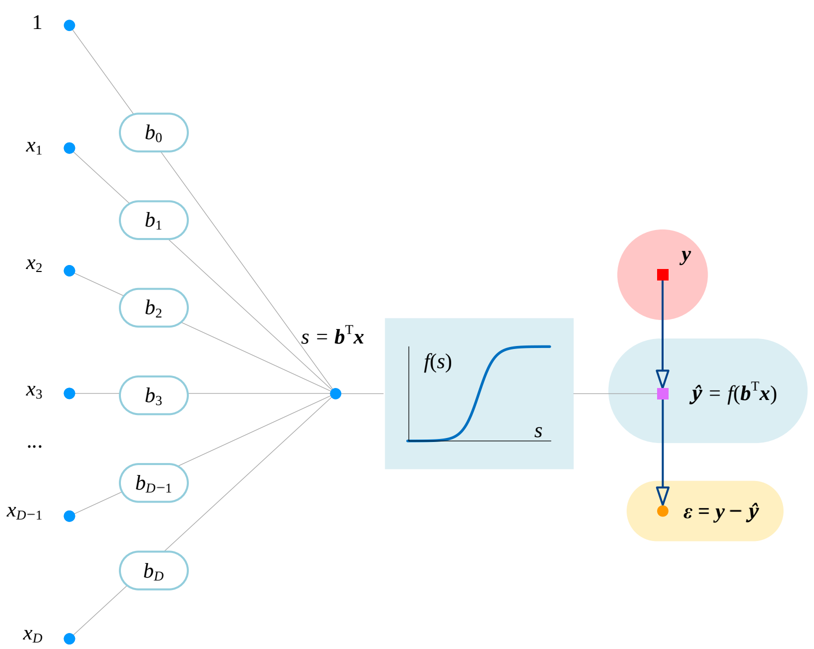


图 25. 逻辑回归和线性回归之间关系

概率视角

形似 (14) 是逻辑分布的 CDF 曲线，对应的表达式：

$$F\left(x|\mu,s\right)=\frac{1}{1+\exp\left(\frac{-\left(x-\mu\right)}{s}\right)}=\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{x-\mu}{2s}\right) \tag{23}$$

其中，μ 为位置参数，s 为形状参数。注意，对于逻辑分布，$s>0$。

逻辑回归可以用来解决二分类，标签为 0 或 1；这是因为逻辑回归可以用来估计事件发生的可能性。

标签为 1 对应的概率为：

$$\Pr\left(y=1|x\right)=\frac{1}{1+\exp\left(-\left(b_0+b_1x\right)\right)} \tag{24}$$

标签为 0 对应的概率为：

$$\Pr\left(y=0|x\right)=1-\Pr\left(y=1|x\right)=\frac{\exp\left(-\left(b_0+b_1x\right)\right)}{1+\exp\left(-\left(b_0+b_1x\right)\right)} \tag{25}$$

图 26 所示为标签为 1 和为 0 的概率关系。

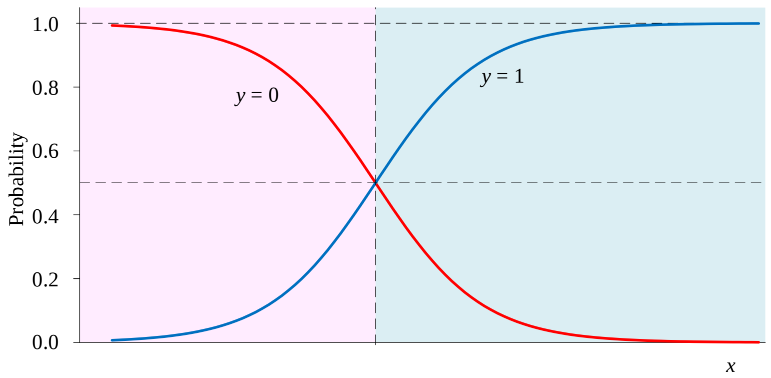


图 26. 标签为 1 和为 0 的概率关系

显然，对于二分类问题，对于任意一点 x，标签为 1 的概率和标签为 0 的概率相加为 1：

$$\mathrm{P}\left(y=0|x\right)+\mathrm{P}\left(y=1|x\right)=1 \tag{26}$$

白话说，某一点要么标签为 1，要么标签为 0，如图 27 所示。

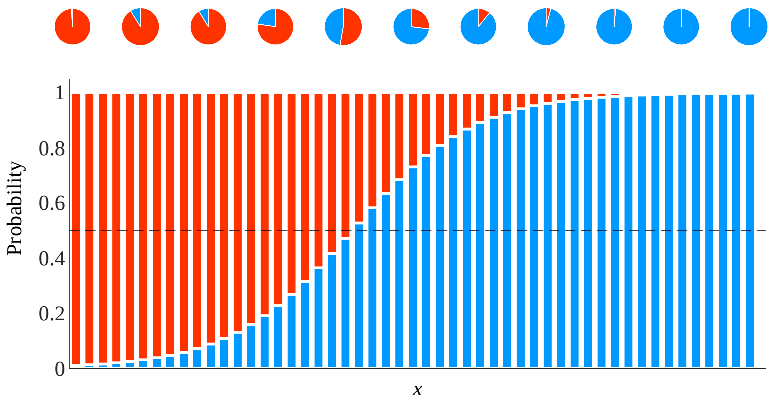


图 27. 逻辑回归模型用于二分类问题

优势率 (odds ratio, OR)，比值比；缩写词为 OR 的对数值：

$$\text{OR} = \text{odds ratio} = \frac{\Pr(y=1|x)}{\Pr(y=0|x)} = \frac{1}{\exp(-(b_0 + b_1 x))} \tag{27}$$

分界 OR = 1，两者概率相同：

$$\frac{1}{\exp(-(b_0 + b_1 x))} = 1 \tag{28}$$

整理得到：

$$b_0 + b_1 x = 0 \tag{29}$$

即

$$x = -\frac{b_0}{b_1} \tag{30}$$

本章后文介绍如何用 sklearn 中逻辑回归函数解决三分类问题。

4.6 逻辑函数完成分类问题

单特征

本节介绍用 sklearn.linear_model.LogisticRegression() 逻辑回归模型，根据鸢尾花花萼长度这一单一特征数据进行分类。

图 28 所示为鸢尾花花萼长度数据和真实三分类 y 之间关系。

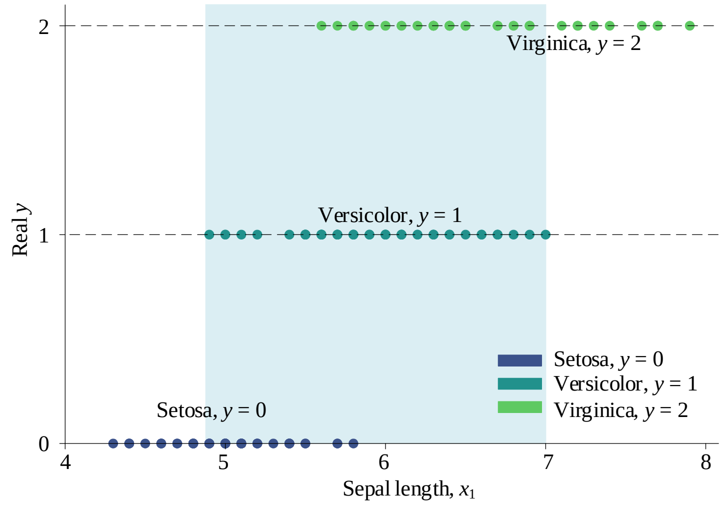


图 28. 鸢尾花花萼长度和真实分类之间关系

图 29 所示为鸢尾花花萼长度数据分类概率密度估计。这幅图实际上已经能够透露出比较合适的分类区间。

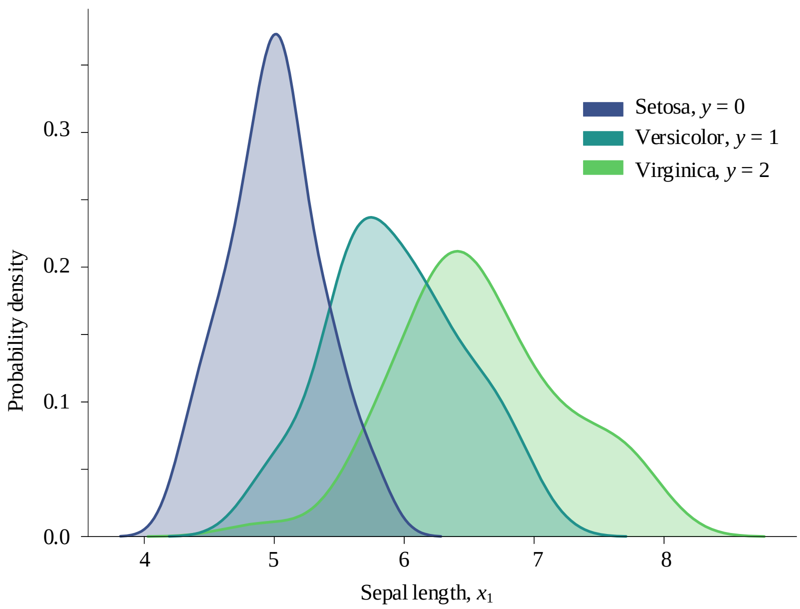


图 29. 鸢尾花花萼长度数据分类概率密度估计

sklearn.linear_model.LogisticRegression() 模型结果可以输出各个分类的概率，得到的图像如图 30 所示。比较三个类别的概率，可以进行分类预测。

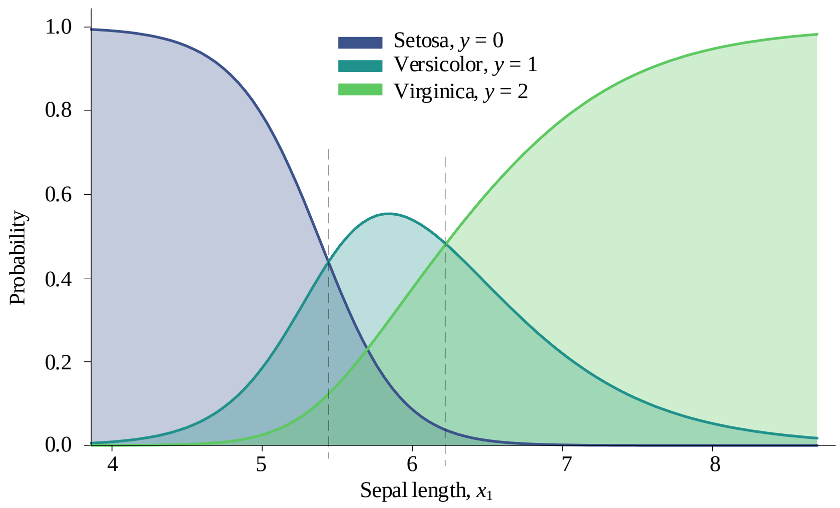


图 30. 逻辑回归估算得到的分类概率

图 31 所示为鸢尾花分类预测结果。

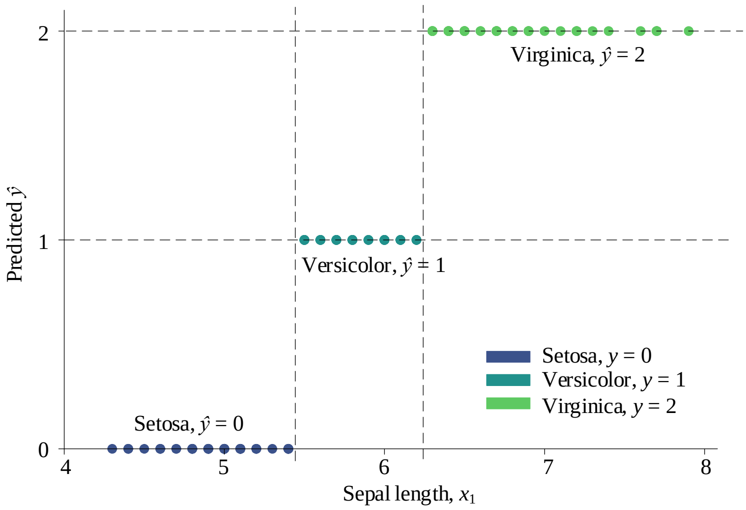


图 31. 鸢尾花花萼长度和预测分类之间关系

Bk6_Ch13_04.py 绘制本节图像。

双特征

本节介绍用 sklearn.linear_model.LogisticRegression() 逻辑回归模型，根据鸢尾花花萼长度和花萼宽度这两个特征数据进行分类。

图 32 所示为鸢尾花花萼长度和花萼宽度两个特征数据散点图，和分类边际分布概率密度估计曲线。

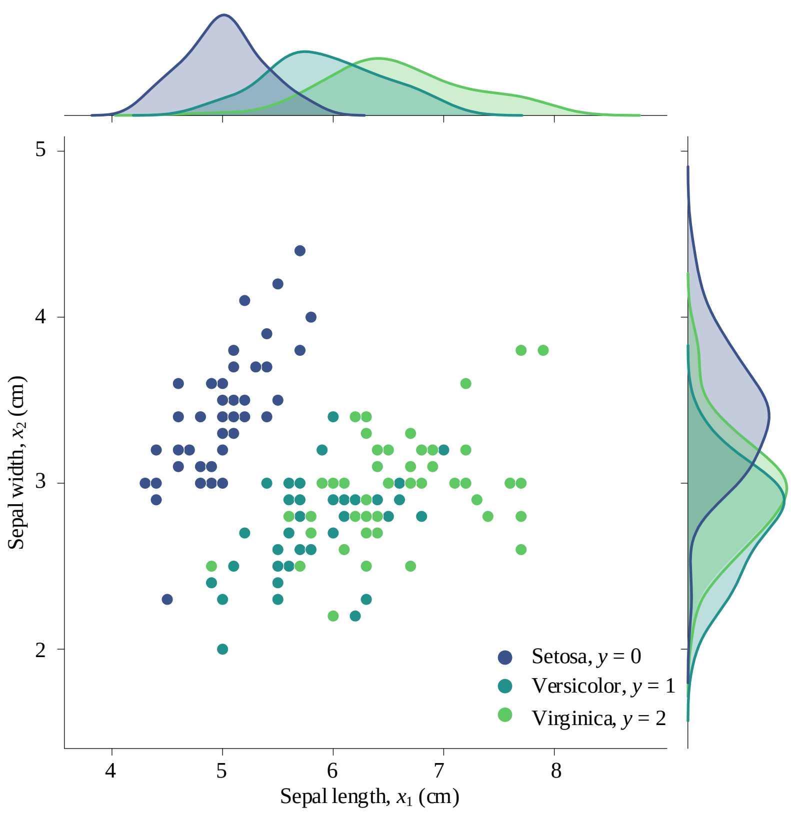


图 32. 鸢尾花双特征数据和分类边际分布

图 33 ~ 图 35 三幅图分别给出鸢尾花双特征分类概率预测曲面。比较三个曲面高度可以得到分类决策边界。在分类问题中，决策边界 (decision boundary) 指的是将不同类别样本分开的平面或曲面。

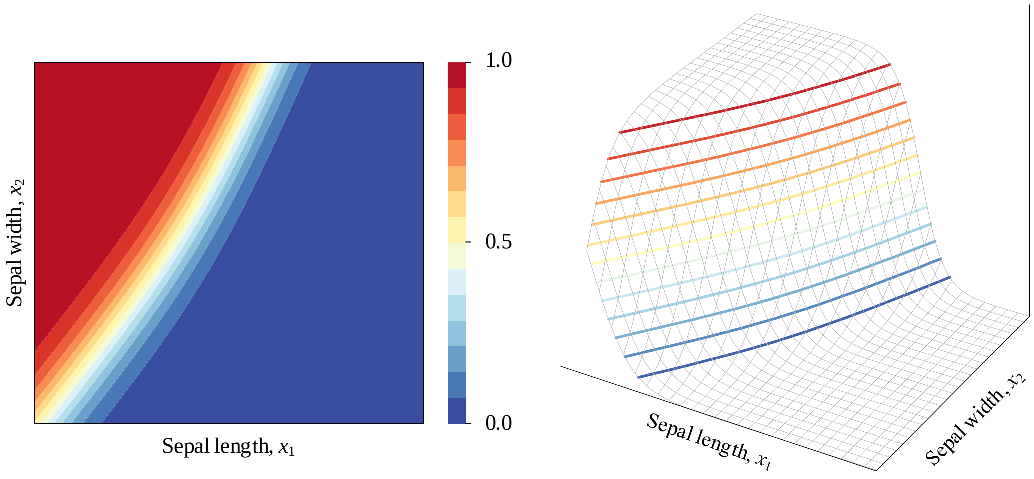


图 33. 鸢尾花双特征分类预测，$\hat{y} = 0$

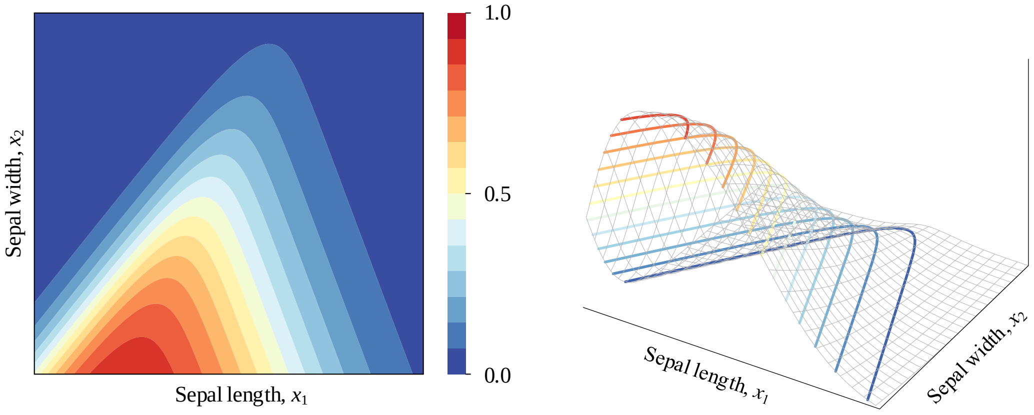


图 34. 鸢尾花双特征分类预测，$\hat{y} = 1$

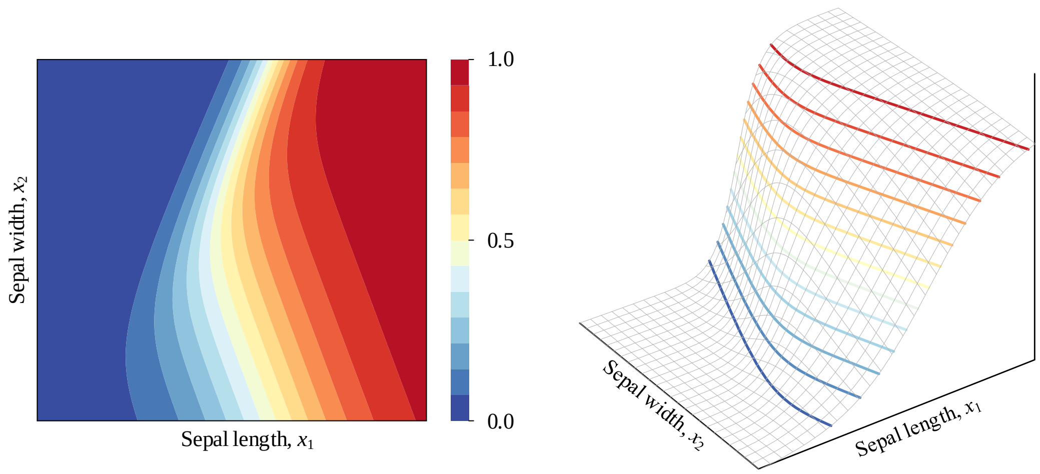


图 35. 鸢尾花双特征分类预测，$\hat{y} = 2$

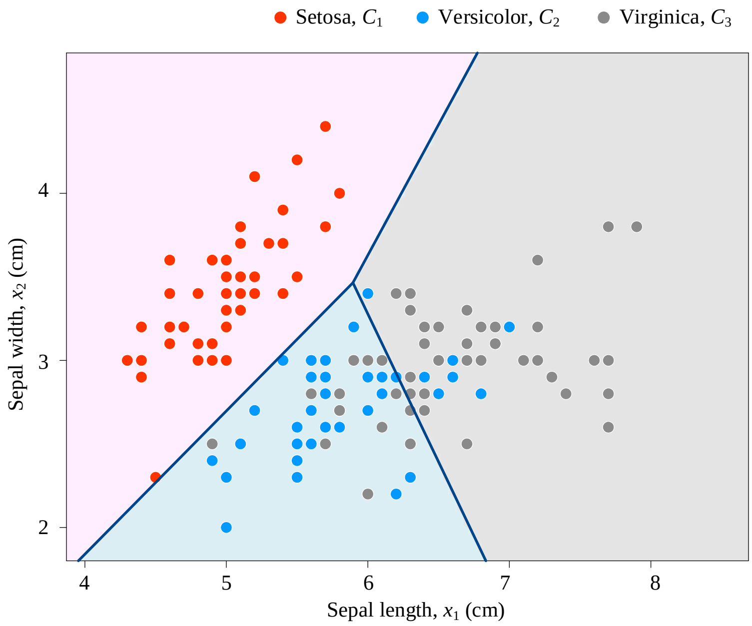


图 36. 利用逻辑回归得到的分类决策边界

Bk6_Ch13_05.py 绘制本节图像。

非线性回归是一种用于建模非线性关系的统计方法。在非线性回归中，因变量和自变量之间的关系不是线性的，而是可以通过非线性函数来描述。

需要非线性回归的原因是许多自然现象和实际问题都不是线性的，例如，随着时间的推移，人口增长率和经济增长率并不是线性的，这就需要非线性回归模型。

常见的非线性回归方法包括多项式回归、指数回归、对数回归、幂函数回归、逻辑回归、等等。每种方法都有其优缺点，例如多项式回归可以拟合大部分的非线性关系，但容易出现过拟合。

逻辑回归将自变量和因变量之间的关系建模为一种逻辑函数，如 sigmoid 函数。从概率视角来看，逻辑回归可以将输出解释为给定输入的条件下，观察到给定类别的概率。它将自变量映射到一个概率值，该值介于 0 和 1 之间，并使用这个概率来预测分类结果。

欢迎读者阅读 *An Introduction to Statistical Learning: With Applications in R* 一书第七章，图书下载地址。

https://www.statlearning.com/