

第14课：逻辑回归——非线性逻辑函数的由来

### 逻辑回归

今天我们要讲的模型叫做 Logistic Regression （LR），一般翻译为逻辑回归。

LR 是一种简单、高效的常用分类模型——有点奇怪是吧，为什么名字叫做“回归”却是一个分类模型，这个我们稍后再讲。先来看看这个 LR 本身。

LR 的模型函数记作：y=h(x)y=h(x)，具体形式如下：

hθ(x)=11+e−θTxhθ(x)=11+e−θTx

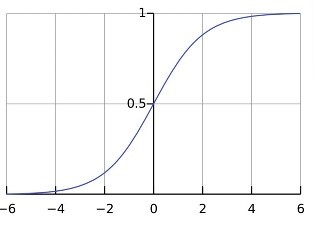
对应到一元自变量的形式为：

h(x)=11+e−(a+bx)h(x)=11+e−(a+bx)

设 z=a+bxz=a+bx，则：

h(z)=11+e−zh(z)=11+e−z

这样的一个函数被称为逻辑函数，它在二维坐标中的表现是这样的：



因为表现为 S 形曲线，所以逻辑函数又被称为 Sigmoid 函数（S 函数）。

Sigmoid 这样一个奇怪而别扭的形式到底是谁、因为什么想出来的呢？又怎么想到用它来做分类的呢？

罗马不是一天建成的，Sigmoid 函数也不是一天形成的。

说起来，逻辑回归的历史相当悠久，迄今大概已经有200年。它的前身，则在18世纪就已经出现了。

18世纪，随着工业革命的深入；世界经济、科技的发展；美洲的发现，以及随之而来的大移民和北美人口迅猛增长……各个学科对于统计学的工具性需求越来越强烈。

到了19世纪，为了研究人口增长以及化学催化反应与时间的关系，人们发明了逻辑函数。

### 指数增长

最初，学者们将人口（或者化合物）数量与时间的函数定义为 W(t)W(t) —— 其中自变量 tt 表示时间， W(t)W(t) 表示人口/化合物数量。

W(t)W(t) 表示的是存量，则 W(t)W(t) 对 tt 求微分的结果 W′(t)W′(t) 就代表了人口/化合物的增长率。

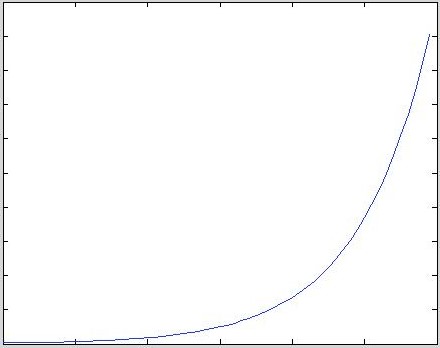
因此增长率函数 W′(t)=dW(t)dtW′(t)=dW(t)dt。

最简单的假设是：W′(t)W′(t) 与 W(t)W(t) 成正比，也就是有：W′(t)=bW(t)W′(t)=bW(t)。

我们知道指数函数 f(x)=exf(x)=ex 的微分函数还是 exex。因此，很自然地，将指数函数引入其中，则有：

W(t)=aebtW(t)=aebt

注意：有时，aa 也可以替换成初始值 W(0)W(0)。它在二维坐标中的表现大致如下：



用 W(t)=aebtW(t)=aebt 去为一个新生国家（比如美国）的人口增长建模，在该国历史的早期被证明是可以有效描述那段时间内的人口增长状况的。

****马尔萨斯****的****人口论****中“在没有任何外界阻碍的情况下，人口将以几何级数增长”的说法，正是依据这样一个模型。

### 逻辑函数

#### 对指数增长的修正

到了19世纪早期，开始有数学家、统计学家质疑 W(t)=aebtW(t)=aebt 模型——任何事物，如果真的按照几何级数任意增长下去，都会达到不可思议的数量。

但是，根据我们的经验，在自然界中，并没有什么东西是在毫无休止地增长的。当一种东西数量越来越多以后，某种阻力也会越来越明显地抑制其增长率。

比利时数学家 Verhulst 给出了一个新的模型：

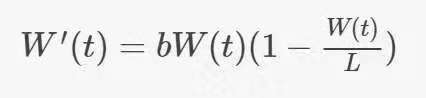
W′(t)=bW(t)−g(W(t))W′(t)=bW(t)−g(W(t))

其中，g(W(t))g(W(t)) 是以 W(t)W(t) 为自变量的函数，它代表随着总数增长出现的阻力。

#### 修正项的具体形式

Verhulst 实验了几种不同形式的阻力（Resistance）函数，当阻力函数表现为 W(t)W(t) 的二次形式时，新模型显示出了它的逻辑性。

因此，新模型的形式可以表示为：



其中 LL 表示 W(t)W(t) 的上限。

这样，增长就与总量、目前总量与最终上限的差 L–W(t)L–W(t) 两者成比例了。

我们设：

P(t)=W(t)LP(t)=W(t)L

则有：

P′(t)=bP(t)(1–P(t))P′(t)=bP(t)(1–P(t))

上式是一个一阶自治微分方程（Autonomous Differential Equations)，我们求解这个微分方程，得出：

P(t)=e(a+bt)1+e(a+bt)P(t)=e(a+bt)1+e(a+bt)

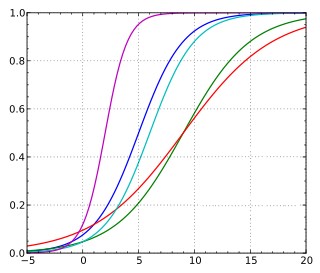
也就是：

P(t)=11+e−(a+bt)P(t)=11+e−(a+bt)

将 tt 替换为 xx，PP 替换为 hh 就是：

h(x)=11+e−(a+bx)h(x)=11+e−(a+bx)

这也是逻辑函数，在二维坐标中的形式如下：



用逻辑函数来描述人口增长的情况，直观上：起初阶段大致是指数增长，随着总量增长，逐渐趋于饱和，增加变慢；最后，达到成熟时增长趋停。

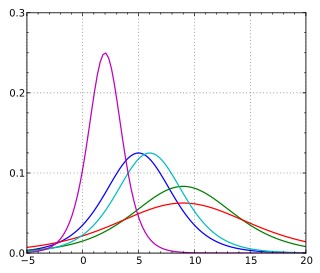
#### 逻辑分布

请注意，逻辑函数表示的是****存量****随时间增长渐增的关系。

而增长率与时间的关系，则是前面讲过的，是存量（逻辑函数）的微分函数：

IMG_261

它的图形是：



从此图更容易看出增长率先变高，后变低的规律。

而且，从形状上看，逻辑分布函数的曲线，还有点像正态分布的倒钟形曲线——实际应用中，逻辑分布在某些领域和场合，曾经被用来作为正态分布的替代。

#### 逻辑函数的通用形式

上面两幅图反映的都是 xx 为一维的情况。当 xx 为多维时，a+bxa+bx 用两个向量相乘—— θTxθTx ——表示，于是逻辑函数就有了如下形式：

hθ(x)=11+e−θTxhθ(x)=11+e−θTx

### 追本溯源的理论学习

上面我们讲了 LR 函数的历史，一方面是为了让大家追随 LR 函数从最初的指数函数形式逐步发展到今天的历程，在追本溯源的过程中由简入繁掌握函数的形式和意义。

另一方面，也是以 LR 函数的形成作为实际案例，来学习****借助数学工具解决问题的方法****：

首先，将目标问题定义为一个函数；

之后，选取最简单的假设作为其具体形式；

然后，用事实数据验证该形式，确认有效后沿用，形成数学模型；

一旦当前采用的数学模型出现问题，则对其进行修正（添加修正项），同样借助事实数据来求取修正项的形式，形成新的（升级版）数学模型。

****直接、简单、基于现有成果****——这也是人类在现实当中解决问题的各种有效方法的共同特征。

我们现在新面临的问题，有些此时尚未得到理想的 Solution。它们最终的解决，也必然遵循“直接、简单、基于现有成果”的原则。

“直接、简单”还比较抽象，但“现有成果”却是看得见摸得着的。

虽然随着新技术的发展，具体的新问题不断涌现，但实际上，技术发展的过程，整体而言是把问题的“量”扩大化的过程。

如果抽象层次更高一些，我们就不难发现，这些问题其实和以往的问题有许多共性所在。其实，新问题不过是旧问题在新情况、新场景下的变形而已。

既然如此，那些已经在实践中证明对于旧问题有效的方法、措施，也必然能够对解决新问题有所贡献。

还有，当一个方法应用到实践中去之后，在最初阶段有效是一种经验，随着经验的积累，研究者的深入探索，经验是有可能被提升为理论的。

理论相对于经验，更加清晰可描述、可解释；抽象层次更高，通用性更强，可以被应用到更广泛的领域。****学习理论相对于学习经验具备更长久也更深入的有效性。****

这也就是时至今日我们还要学习几千年前的几何定理、几百年前的物理化学定律的缘故。

机器学习就是这样一套已经逐步在形成理论的方法。相比而言，更新的深度学习还处在经验阶段。

尽管今时今日从媒体传播的角度来看，深度学习是一个更热门的领域，对比起来，似乎机器学习“过时了”，“没用了”。

但在解决人们的实际问题上，机器学习的应用远比深度学习广泛、深入、可控。

而且，机器学习是深度学习的基础。如果连最简单的机器学习模型都没有掌握，又怎么可能了解深度神经网络的原理和运作？

如果连 How it works 都不知道，就拿来当黑盒使用，那么无论是什么工具，你对它的应用就变成了“按一下电钮”那么简单，你个人的价值又如何体现呢？

在此加入这样一个小节，是希望大家能够借助实例认识到理论学习的意义和作用，从而端正学习态度。

### 线性 VS 非线性

对照我们之前讲过的线性函数的定义，可知：****逻辑函数是非线性函数****。

自变量与因变量的线性关系和非线性关系，到底到怎么理解呢？

最直观的理解，就是将自变量和因变量带入对应维度的坐标系中，在对应值“描点”，然后看看这些点组成的图形。

线性：二维坐标系中的直线，三维坐标系中的平面……

非线性：二位坐标中的曲线（严格的来讲，直线也是一种特殊的曲线，但为了方便而言，我们在此处用“曲”来指代“非直”。“非直”包括“弯曲”，也包括 ReLU 函数这种“一段段拼接的线段”）；三维坐标中的曲面……

换一个角度而言，****线性关系表达的是一种相关性****。

根据线性关系，通过既往数据获知：不同的事物在多大程度上可能共同发生。以此为依据来判断：未来，在某些事物（自变量）发生后，另一些事物（因变量）是否会发生。

线性回归因其简单、容易理解、计算量低等特点，而在现实中得到了广泛应用（基本上，任何事情，上来先做个线性回归，即使无法得出结论，也不会消耗太多资源）。

但是，相关不等于因果。两件事会同时发生，并不是说它们之间就有因果关系。还有许许多多的情况无法用简单的相关来模型化。在这些情况下，线性模型往往无法揭示自变量和因变量的关系。

由于线性模型的明显局限，许多研究者把焦点转向了非线性模型。不过非线性模型对于计算量的需求远大于线性模型。对于资源的大量需求在一定程度上限制了非线性模型的应用。

有时候，一个问题用线性模型，能够得到一个七八十分、“尚且可用”的解决方案；应用非线性模型则能得到九十分，甚至九十五分的优质解决方案。但后者耗费过于巨大，导致你无法承受。很可能你宁可选一个还能用但“便宜”的，去追求性价比，而不是追求高质量。

当然，随着计算机硬件和分布式系统、分布式计算的发展，运算能力也在飞速发展。相应的，非线性模型的应用场景也越来越多。不过质量和资源的矛盾总是存在的，这一点在我们处理实际问题的时候，总是需要考虑。

[邀请好友一起学，获得 25% 返现奖励](https://gitbook.cn/m/mazi/columns/5bc6ac7442d7d32f50f19a98/topics/5bf2821dfd72950cafdccd90)

[IMG_263](https://gitbook.cn/m/mazi/comp/column?columnId=5bc6ac7442d7d32f50f19a98%26tag=2#catalog)