

第二章 概率与信息论

概率论是用于表示不确定性声明的数学框架。它不仅提供了量化不确定性的方法，也提供了用于导出新的不确定性**声明**（statement）的公理。在人工智能领域，概率论主要有两种用途。首先，概率法则告诉我们 AI 系统**如何推理**，据此我们设计一些算法来计算或者估算由概率论导出的表达式。其次，我们可以用概率和统计从理论上分析我们提出的 AI **系统的行为**。

概率论使我们能够提出**不确定的声明**以及在不确定性存在的情况下进行**推理**，而信息论使我们能够量化概率分布中的**不确定性总量**。

2.1 为什么要使用概率？

不确定性有三种可能的来源：

1、被建模系统**内在的随机性**。例如，大多数量子力学的解释，都将亚原子粒子的动力学描述为概率的。我们还可以创建一些我们假设具有随机动态的理论情境，例如一个假想的纸牌游戏，在这个游戏中我们假设纸牌被真正混洗成了随机顺序。

2、**不完全观测**。即使是确定的系统，当我们不能观测到所有驱动系统行为的变量时，该系统也会呈现随机性。例如，在 Monty Hall 问题中，一个游戏节目的参与者被要求在三个门之间选择，并且会赢得放置在选中门后的奖品。其中两扇门通向山羊，第三扇门通向一辆汽车。选手的每个选择所导致的结果是确定的，但是站在选手的角度，结果是不确定的。

3、**不完全建模**。当我们使用一些必须舍弃某些观测信息的模型时，舍弃的信息会导致模型的预测出现不确定性。例如，假设我们制作了一个机器人，它可以准确地观察周围每一个对象的位置。在对这些对象将来的位置进行预测时，如果机器人采用的是离散化的空间，那么离散化的方法将使得机器人无法确定对象们的精确位置：因为每个对象都可能处于它被观测到的离散单元的任何一个角落。

2.2 随机变量

随机变量（random variable）是可以随机地取不同值的变量。我们通常用无格式字体(plain typeface)中的小写字母来表示随机变量本身，而用手写体中的小写字母来表示

随机变量能够取到的值。例如， x_1 和 x_2 都是随机变量 x 可能的取值。对于向量值变量，我们会将随机变量写成 \mathbf{x} ，它的一个可能取值为 \mathbf{x} 。就其本身而言，一个随机变量只是对可能的状态的描述；它必须伴随着一个概率分布来指定每个状态的可能性。

随机变量可以是离散的或者连续的。离散随机变量拥有有限或者可数无限多的状态。注意这些状态不一定非要是整数；它们也可能只是一些被命名的状态而没有数值。连续随机变量伴随着实数值。

2.3 概率分布

概率分布 (probability distribution) 用来描述随机变量或一簇随机变量在每一个可能取到的状态的可能性大小。我们描述概率分布的方式取决于随机变量是离散的还是连续的。

2.3.1 离散型变量和概率质量函数

离散型变量的概率分布可以用概率质量函数 (probability mass function, PMF)¹ 来描述。我们通常用大写字母 P 来表示概率质量函数。通常每一个随机变量都会有一个不同的概率质量函数，并且读者必须根据随机变量来推断所使用的 PMF，而不是根据函数的名称来推断；例如， $P(x)$ 通常和 $P(y)$ 不一样。

概率质量函数将随机变量能够取得的每个状态映射到随机变量取得该状态的概率。 $x=x$ 的概率用 $P(x)$ 来表示，概率为 1 表示 $x=x$ 是确定的，概率为 0 表示 $x=x$ 是不可能发生的。有时为了使得 PMF 的使用不相互混淆，我们会明确写出随机变量的名称： $P(x=x)$ 。有时我们会先定义一个随机变量，然后用“ \sim ”符号来说明它遵循的分布： $x \sim P(x)$ 。

概率质量函数可以同时作用于多个随机变量。这种多个变量的概率分布被称为联合概率分布 (joint probability distribution)。 $P(x=x, y=y)$ 表示 $x=x$ 和 $y=y$ 同时发生的概率。我们也可以简写为 $P(x, y)$ 。

如果一个函数 P 是随机变量 x 的 PMF，必须满足下面这几个条件：

- P 的定义域必须是 x 所有可能状态的集合。
- $\forall x \in x, 0 \leq P(x) \leq 1$ 。不可能发生的事件概率为 0，并且不存在比这概率更低的状态。类似的，能够确保一定发生的事件概率为 1，并且不存在比这概率更高的状态。

¹ 也有教材的翻译成概率分布律。

- $\sum_{x \in \mathcal{X}} P(x) = 1$ 。我们把这条性质称之为归一化的 (normalized)。如果没有这条性质，当我们计算很多事件其中之一发生的概率时可能会得到大于 1 的概率。

例如，考虑一个离散型随机变量 x 有 k 个不同的状态。我们可以假设 x 是均匀分布 (uniform distribution) 的 (也就是将它的每个状态视为等可能的)，通过将它的 PMF 设为

$$P(x=x_i) = \frac{1}{k}$$

对于所有的 i 都成立。我们可以看出这满足上述成为概率质量函数的条件。因为 k 是一个正整数，所以 $\frac{1}{k}$ 是正的。我们也可以看出

$$\sum_i P(x=x_i) = \sum_i \frac{1}{k} = \frac{k}{k} = 1$$

因此分布也满足归一化条件。

2.3.2 连续型变量和概率密度函数

当我们研究的对象是连续型随机变量时，我们用概率密度函数 (probability density function, PDF) 而不是概率质量函数来描述它的概率分布。如果一个函数 p 是概率密度函数，必须满足下面这几个条件：

- p 的定义域必须是 x 所有可能状态的集合。
- $\forall x \in \mathcal{X}, p(x) \geq 0$ 。注意，我们并不要求 $p(x) \leq 1$ 。
- $\int p(x) dx = 1$ 。

概率密度函数 $p(x)$ 并没有直接对特定的状态给出概率，相对的，它给出了落在面积为 δx 的无限小的区域内的概率为 $p(x)\delta x$ 。

我们可以对概率密度函数求积分来获得点集的真实概率质量。特别地， x 落在集合 S 中的概率可以通过 $p(x)$ 对这个集合求积分来得到。在单变量的例子中， x 落在区间 $[a, b]$ 的概率是 $\int_{[a, b]} p(x) dx$ 。

为了给出一个连续型随机变量的 PDF 的例子，我们可以考虑实数区间上的均匀分

布。我们可以使用函数 $u(x; a, b)$ ，其中 a 和 b 是区间的端点且满足 $b > a$ 。符号 “;” 表示“以什么为参数”；我们把 x 作为函数的自变量， a 和 b 作为定义函数的参数。为了确保区间外没有概率，我们对所有的 $x \notin [a, b]$ ，令 $u(x; a, b) = 0$ 。在 $[a, b]$ 内，有 $u(x; a, b) = \frac{1}{b-a}$ 。我们可以看出任何一点都非负。另外，它的积分为 1。我们通常用 $x \sim U(a, b)$ 表示 x 在 $[a, b]$ 上是均匀分布的。

2.4 边缘概率

有时候，我们知道了一组变量的联合概率分布，但想要了解其中一个子集的概率分布。这种定义在子集上的概率分布被称为边缘概率分布 (marginal probability distribution)。

例如，假设有离散型随机变量 x 和 y ，并且我们知道 $P(x, y)$ 。我们可以依据下面的求和法则 (sum rule) 来计算 $P(x)$ ：

$$\forall x \in \mathcal{X}, P(x=x) = \sum_y P(x=x, y=y)$$

“边缘概率”的名称来源于手算边缘概率的计算过程。当 $P(x, y)$ 的每个值被写在由每行表示不同的 x 值，每列表示不同的 y 值形成的网格中时，对网格中的每行求和是很自然的事情，然后将求和的结果 $P(x)$ 写在每行右边的纸的边缘处。

对于连续型变量，我们需要用积分替代求和：

$$p(x) = \int p(x, y) dy$$

2.5 条件概率

在很多情况下，我们感兴趣的是某个事件，在给定其他事件发生时出现的概率。这种概率叫做条件概率。我们将给定 $x=x, y=y$ 发生的条件概率记为 $P(y=y | x=x)$ 。这个条件概率可以通过下面的公式计算：

$$P(y=y | x=x) = \frac{P(y=y, x=x)}{P(x=x)}$$

条件概率只在 $P(x=x) > 0$ 时有定义。我们不能计算给定在永远不会发生的事件上的条件概率。

2.6 条件概率的链式法则

任何多维随机变量的联合概率分布，都可以分解成只有一个变量的条件概率相乘的形式：

$$P(x^{(1)}, \dots, x^{(n)}) = P(x^{(1)}) \prod_{i=2}^n P(x^{(i)} | x^{(1)}, \dots, x^{(i-1)})$$

这个规则被称为概率的链式法则 (chain rule) 或者乘法法则 (product rule)。例如，使用两次定义可以得到

$$\begin{aligned} P(a, b, c) &= P(a | b, c) P(b, c) \\ P(b, c) &= P(b | c) P(c) \\ \Rightarrow P(a, b, c) &= P(a | b, c) P(b | c) P(c) \end{aligned}$$

2.7 独立性和条件独立性

两个随机变量 x 和 y ，如果它们的概率分布可以表示成两个因子的乘积形式，并且一个因子只包含 x 另一个因子只包含 y ，我们就称这两个随机变量是相互独立的 (independent)：

$$\forall x \in \mathbf{x}, y \in \mathbf{y}, p(x = x, y = y) = p(x = x) p(y = y)$$

如果关于 x 和 y 的条件概率分布对于 z 的每一个值都可以写成乘积的形式，那么这两个随机变量 x 和 y 在给定随机变量 z 时是条件独立的 (conditionally independent)：

$$\forall x \in \mathbf{x}, y \in \mathbf{y}, z \in \mathbf{z}, p(x = x, y = y | z = z) = p(x = x | z = z) p(y = y | z = z)$$

我们可以采用一种简化形式来表示独立性和条件独立性： $x \perp y$ 表示 x 和 y 相互独立， $x \perp y | z$ 表示 x 和 y 在给定 z 时条件独立。

2.8 期望、方差和协方差

函数 $f(x)$ 关于某分布 $P(x)$ 的期望 (expectation) 或者期望值 (expected value) 是指，当 x 由 P 产生， f 作用于 x 时， $f(x)$ 的平均值。对于离散型随机变量，这可以通过求和

得到：

$$\mathbb{E}_{x \sim P}[f(x)] = \sum_x P(x)f(x)$$

对于连续型随机变量可以通过求积分得到：

$$\mathbb{E}_{x \sim p}[f(x)] = \int p(x)f(x)dx$$

期望是线性的，例如，

$$\mathbb{E}_x[\alpha f(x) + \beta g(x)] = \alpha \mathbb{E}_x[f(x)] + \beta \mathbb{E}_x[g(x)]$$

其中 α 和 β 不依赖于 x 。

方差 (variance) 衡量的是当我们对 x 依据它的概率分布进行采样时，随机变量 x 的函数值会呈现多大的差异：

$$\text{Var}(f(x)) = \mathbb{E}\left[\left(f(x) - \mathbb{E}[f(x)]\right)^2\right]$$

当方差很小时， $f(x)$ 的值形成的簇比较接近它们的期望值。方差的平方根被称为**标准差** (standard deviation)。

协方差 (covariance) 在某种意义上给出了两个**变量线性相关性的强度**以及这些变量的尺度：

$$\text{Cov}(f(x), g(y)) = \mathbb{E}\left[\left(f(x) - \mathbb{E}[f(x)]\right)\left(g(y) - \mathbb{E}[g(y)]\right)\right]$$

协方差的绝对值如果很大则意味着变量值变化很大并且它们同时距离各自的均值很远。如果协方差是正的，那么两个变量都倾向于同时**取得相对较大的值**。如果协方差是负的，那么其中一个变量倾向于取得相对较大的值的同时，另一个变量倾向于取得相对较小的值，反之亦然。其他的衡量指标如**相关系数** (correlation) 将每个变量的贡献归一化，为了只衡量变量的相关性而不受各个变量尺度大小的影响。

协方差和相关性是有联系的，但实际上是不同的概念。它们是有联系的，因为两个变量如果**相互独立那么它们的协方差为零**，如果两个变量的协方差不为零那么它们一定是相关的。然而，独立性又是和协方差完全不同的性质。两个变量如果协方差为零，它们之间**一定没有线性关系**。独立性比零协方差的要求更强，因为**独立性还排除了非线性**

的关系。两个变量相互依赖但具有零协方差是可能的。例如，假设我们首先从区间 $[-1,1]$ 上的均匀分布中采样出一个实数 x 。然后我们对一个随机变量 s 进行采样。 s 以 $\frac{1}{2}$ 的概率值为1，否则为-1。我们可以通过令 $y = sx$ 来生成一个随机变量 y 。显然， x 和 y 不是相互独立的，因为 x 完全决定了 y 的尺度。然而， $\text{Cov}(x, y) = 0$ 。

随机向量 $\mathbf{x} \in \mathbb{R}^n$ 的协方差矩阵（covariance matrix）是一个 $n \times n$ 的矩阵，并且满足

$$\text{Cov}(\mathbf{x})_{i,j} = \text{Cov}(x_i, x_j)$$

协方差矩阵的对角元是方差：

$$\text{Cov}(x_i, x_i) = \text{Var}(x_i)$$

2.9 常用概率分布

2.9.1 Bernoulli 分布（伯努利分布）

Bernoulli 分布（Bernoulli distribution）是单个二值随机变量的分布。它由单个参数 $\phi \in [0,1]$ 控制， ϕ 给出了随机变量等于1的概率。它具有如下的一些性质：

$$\begin{aligned} P(x=1) &= \phi \\ P(x=0) &= 1-\phi \\ p(x=x) &= \phi^x(1-\phi)^{1-x} \\ \mathbb{E}_x[x] &= \phi \\ \text{Var}_x(x) &= \phi(1-\phi) \end{aligned}$$

2.9.2 Multinoulli 分布

Multinoulli 分布（multinoulli distribution）或者范畴分布（categorical distribution）是指在具有 k 个不同状态的单个离散型随机变量上的分布，其中 k 是一个有限值²。

Multinoulli 分布由向量 $\mathbf{p} \in [0,1]^{k-1}$ 参数化，其中每一个分量 p_i 表示第 i 个状态的概率。最后的第 k 个状态的概率可以通过 $1 - \mathbf{1}^T \mathbf{p}$ 给出。注意我们必须限制 $\mathbf{1}^T \mathbf{p} \leq 1$ 。Multinoulli 分布经常用来表示对象分类的分布，所以我们很少假设状态1具有数值1之类的。因此，

²“multinoulli”这个术语是最近被 Gustavo Lacerdo 明、被 Murphy(2012) 推广的。Multinoulli 分布是多项式分布(multinomial distribution)的一个特例。项式分布是 $\{0, \dots, n\}^k$ 中的向量的分布，用于表示当对 Multinoulli 分布采样 n 次时 k 个类中的每一个被访问的次数。很多文章使用“多项式分布”而实际上说的是 Multinoulli 分布，但是他们并没有说是对 $n=1$ 的情况，这点需要注意。

我们通常不需要去计算 Multinoulli 分布的随机变量的期望和方差。

Bernoulli 分布和 Multinoulli 分布足够用来描述在它们领域内的任意分布。它们能够描述这些分布，不是因为它们特别强大，而是因为它们的领域很简单；它们可以对那些，能够将所有的状态进行枚举的离散型随机变量进行建模。当处理的是连续型随机变量时，会有不可数无限多的状态，所以任何通过少量参数描述的概率分布都必须在分布上加以严格的限制。

2.9.3 高斯分布

实数上最常用的分布就是正态分布 (normal distribution)，也称为高斯分布 (Gaussian distribution)：

$$N(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right)$$

画出了正态分布的概率密度函数。

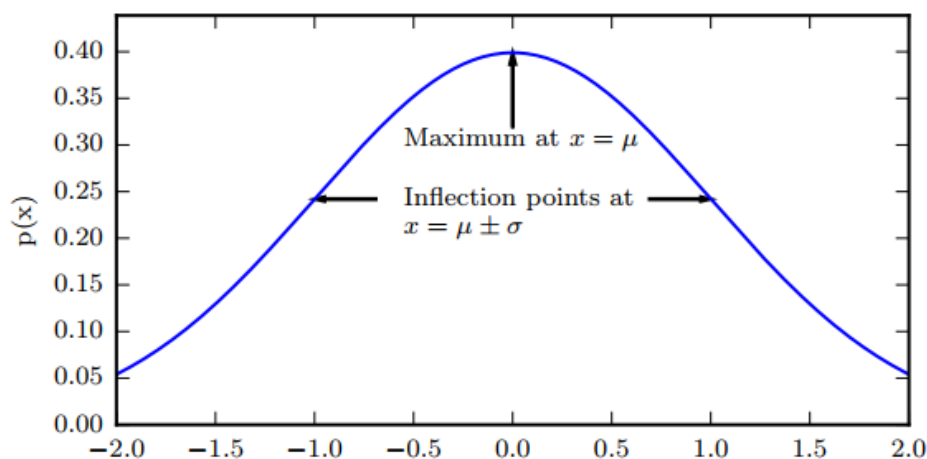


图 17 正态分布。正态分布 $N(x; \mu, \sigma^2)$ 呈现经典的“钟形曲线”的形状，其中中心峰的 x 坐标由 μ 给出，峰的宽度受 σ 控制。在这个示例中，我们展示的是标准正态分布 (standard normal distribution)，其中 $\mu=0; \sigma=1$ 。

正态分布由两个参数控制， $\mu \in \mathbb{R}$ 和 $\sigma \in (0, \infty)$ 。参数 μ 给出了中心峰值的坐标，这也是分布的均值： $\mathbb{E}[x] = \mu$ 。分布的标准差用 σ 表示，方差用 σ^2 表示。

当我们要对概率密度函数求值时，我们需要对 σ 平方并且取倒数。当我们需要经常对不同参数下的概率密度函数求值时，一种更高效的参数化分布的方式是使用参数 $\beta \in (0, \infty)$ ，来控制分布的精度 (precision) (或方差的倒数)：

$$N(x; \mu, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2}(x - \mu)^2\right)$$

采用正态分布在很多应用中都是一个明智的选择。当我们由于缺乏关于某个实数上分布的先验知识而不知道该选择怎样的形式时，正态分布是默认的比较好的选择，其中有两个原因。

第一，我们想要建模的很多分布的真实情况是比较接近正态分布的。中心极限定理（central limit theorem）说明很多独立随机变量的和近似服从正态分布。这意味着在实际中，很多复杂系统都可以被成功地建模成正态分布的噪声，即使系统可以被分解成一些更结构化的部分。

第二，在具有相同方差的所有可能的概率分布中，正态分布在实数上具有最大的不确定性。因此，我们可以认为正态分布是对模型加入的先验知识量最少的分布。

正态分布可以推广到 \mathbb{R}^n 空间，这种情况下被称为多维正态分布（multivariate normal distribution）。它的参数是一个正定对称矩阵 Σ ：

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right)$$

参数 μ 仍然表示分布的均值，只不过现在是向量值。参数 Σ 给出了分布的协方差矩阵。和单变量的情况类似，当我们希望对很多不同参数下的概率密度函数多次求值时，协方差矩阵并不是一个很高效的参数化分布的方式，因为对概率密度函数求值时需要对 Σ 求逆。我们可以使用一个精度矩阵（precision matrix） β 进行替代：

$$N(x; \mu, \beta^{-1}) = \sqrt{\frac{\det(\beta)}{(2\pi)^n}} \exp\left(-\frac{1}{2}(x - \mu)^T \beta(x - \mu)\right)$$

我们常常把协方差矩阵固定成一个对角阵。一个更简单的版本是各向同性（isotropic）高斯分布，它的协方差矩阵是一个标量乘以单位阵。

2.9.4 指数分布和 Laplace 分布

在深度学习中，我们经常会需要一个在 $x=0$ 点处取得边界点(sharp point)的分布。为了实现这一目的，我们可以使用指数分布（exponential distribution）：

$$p(x; \lambda) = \lambda \mathbf{1}_{x \geq 0} \exp(-\lambda x)$$

指数分布使用指示函数(indicator function) $\mathbf{1}_{x \geq 0}$ 来使得 x 当取负值时的概率为零。

一个联系紧密的概率分布是 **Laplace 分布** (Laplace distribution)，它允许我们在任意一点 μ 处设置概率质量的峰。

$$\text{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp\left(-\frac{|x - \mu|}{\gamma}\right)$$

2.9.5 Dirac 分布和经验分布

在一些情况下，我们希望概率分布中的所有质量都集中在一个点上。这可以通过 **Dirac delta 函数** (Dirac delta function) $\delta(x)$ 定义概率密度函数来实现：

$$p(x) = \delta(x - \mu)$$

Dirac delta 函数被定义成在除了 0 以外的所有点的值都为 0，但是积分为 1。Dirac delta 函数不像普通函数一样对 x 的每一个值都有一个实数值的输出，它是一种不同类型的数学对象，被称为 **广义函数** (generalized function)，广义函数是依据积分性质定义的数学对象。我们可以把 Dirac delta 函数想成 **一系列函数的极限点**，这一系列函数把除 0 以外的所有点的概率密度越变越小。

通过把 $p(x)$ 定义成 δ 函数左移 $-\mu$ 个单位，我们得到了一个在 $x = \mu$ 处具有无限窄也无限高的峰值的概率质量。

Dirac 分布经常作为 **经验分布** (empirical distribution) 的一个组成部分出现：

$$\hat{p}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

经验分布将概率密度 $\frac{1}{m}$ 赋给 m 个点 $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}$ 中的每一个，这些点是给定的数据集或者采样的集合。只有在定义连续型随机变量的经验分布时，Dirac delta 函数才是必要的。对于离散型随机变量，情况更加简单：经验分布可以被定义成一个 Multinoulli 分布，对于每一个可能的输入，其概率可以简单地设为在训练集上那个输入值的 **经验频率** (empirical frequency)。

当我们在训练集上训练模型时，我们可以认为从这个训练集上得到的经验分布 **指明了我们采样来源的分布**。关于经验分布另外一种重要的观点是，它是训练数据的 **似然最大** (**就是极大似然估计的直接理解**) 的那个概率密度函数。

2.9.6 分布的混合

略。

2.10 常用函数的有用性质

某些函数在处理概率分布时经常会出现，尤其是深度学习的模型中用到的概率分布。

其中一个函数是 **logistic sigmoid** 函数，在变量取绝对值非常大的正值或负值时会出现**饱和** (saturate) 现象，意味着函数会变得很平，并且对输入的微小改变会变得不敏感。

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

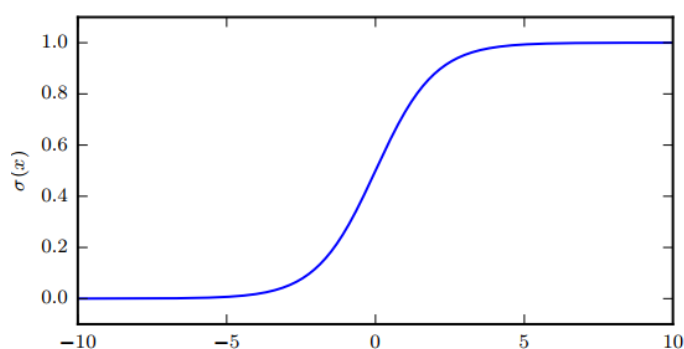


图 18 logistic sigmoid 函数

另外一个经常遇到的函数是 **softplus 函数** (softplus function)

$$\zeta(x) = \log(1 + \exp(x))$$

softplus 函数可以用来产生正态分布的 β 和 σ 参数，因为它的范围是 $(0, \infty)$ 。当处理包含 sigmoid 函数的表达式时它也经常出现。softplus 函数名来源于它是另外一个函数的平滑（或“软化”）形式，这个函数是

$$x^+ = \max(0, x)$$

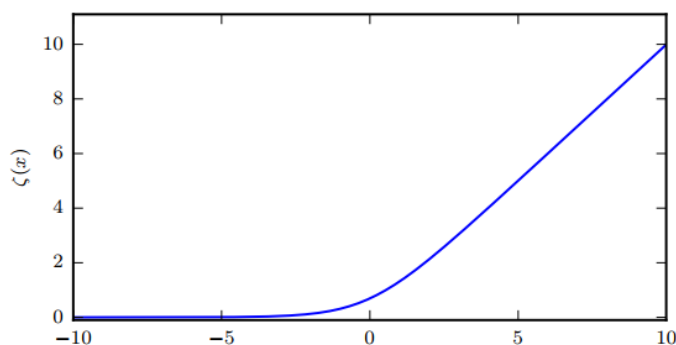


图 19 softplus 函数

下面一些性质非常有用，你可能要记下来：

$$\begin{aligned}\sigma(x) &= \frac{\exp(x)}{\exp(x) + \exp(0)} \\ \frac{d}{dx}\sigma(x) &= \sigma(x)(1 - \sigma(x)) \\ 1 - \sigma(x) &= \sigma(-x) \\ \log \sigma(x) &= -\zeta(-x) \\ \frac{d}{dx}\zeta(x) &= \sigma(x) \\ \forall x \in (0, 1), \sigma^{-1}(x) &= \log\left(\frac{x}{1-x}\right) \\ \forall x > 0, \zeta^{-1}(x) &= \log(\exp(x) - 1) \\ \zeta(x) &= \int_{-\infty}^x \sigma(y) dy \\ \zeta(x) - \zeta(-x) &= x\end{aligned}$$

图 20 函数性质

2.11 贝叶斯规则

我们经常会需要在已知 $P(y|x)$ 时计算 $P(x|y)$ 。幸运的是，如果还知道 $P(x)$ ，我们可以用 **贝叶斯规则**（Bayes' rule）来实现这一目的：

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)}$$

注意到 $P(y)$ 出现在上面的公式中，它通常使用 $P(y) = \sum_x P(y|x)P(x)$ 来计算，所以我们并不需要事先知道 $P(y)$ 的信息。

2.12 连续型变量的技术细节

略。

2.13 信息论

信息论是应用数学的一个分支，主要研究的是**对一个信号包含信息的多少进行量化**。它最初被发明是用来研究在一个含有噪声的信道上用离散的字母表来发送消息，例如通过无线电传输来通信。在这种情况下，信息论告诉我们如何对消息设计最优编码以及计

算消息的期望长度，这些消息是使用多种不同编码机制、从特定的概率分布上采样得到的。在机器学习中，我们也可以把信息论应用于连续型变量，此时某些消息长度的解释不再适用。信息论是电子工程和计算机科学中许多领域的基础。在本书中，我们主要使用信息论的一些关键思想来描述概率分布或者量化概率分布之间的相似性。

信息论的基本想法是一个不太可能的事件居然发生了，要比一个非常可能的事件发生，能提供更多的信息。消息说：“今天早上太阳升起”信息量是如此之少以至于没有必要发送，但一条消息说：“今天早上有日食”信息量就很丰富。

我们想要通过这种基本想法来量化信息。特别地，

- 非常可能发生的事件信息量要比较少，并且极端情况下，确保能够发生的事件应该没有信息量。
- 较不可能发生的事件具有更高的信息量。
- 独立事件应具有增量的信息。例如，投掷的硬币两次正面朝上传递的信息量，应该是投掷一次硬币正面朝上的信息量的两倍。

为了满足上述三个性质，我们定义一个事件 $x = x$ 的自信息（self-information）为

$$I(x) = -\log P(x)$$

在本书中，我们总是用 \log 来表示自然对数，其底数为 e 。因此我们定义的 $I(x)$ 单位是奈特（nats）。一奈特是以 $\frac{1}{e}$ 的概率观测到一个事件时获得的信息量。

自信息只处理单个的输出。我们可以用香农熵（Shannon entropy）来对整个概率分布中的不确定性总量进行量化：

$$H(x) = \mathbb{E}_{x \sim P} [I(x)] = -\mathbb{E}_{x \sim P} [\log P(x)]$$

一个分布的香农熵是指遵循这个分布的事件所产生的期望信息总量。它给出了对依据概率分布 P 生成的符号进行编码所需的比特数在平均意义上的下界(当对数底数不是2时，单位将有所不同)。那些接近确定性的分布(输出几乎可以确定)具有较低的熵；那些接近均匀分布的概率分布具有较高的熵。图 21 给出了一个说明。当 x 是连续的，香农熵被称为微分熵（differential entropy）。

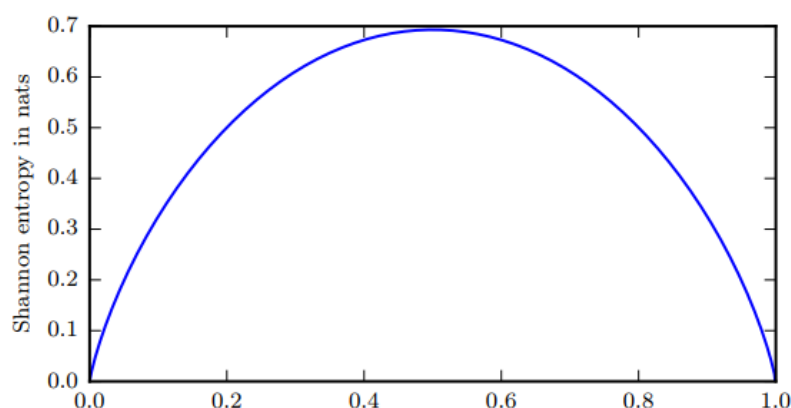


图 21 二值随机变量的香农熵。该图说明了更接近确定性的分布是如何具有较低的香农熵，而更接近均匀分布的分布是如何具有较高的香农熵。水平轴是 p ，表示二值随机变量等于 1 的概率。熵由 $(p-1)\log(1-p)-p\log p$ 给出。当 p 接近 0 时，分布几乎是确定的，因为随机变量几乎总是 0。当 p 接近 1 时，分布也几乎是确定的，因为随机变量几乎总是 1。当 $p=0.5$ 时，熵是最大的，因为分布在两个结果（0 和 1）上是均匀的。

如果我们对于同一个随机变量 x 有两个单独的概率分布 $P(x)$ 和 $Q(x)$ ，我们可以使用 **KL 散度**（Kullback-Leibler (KL) divergence）来衡量这两个分布的差异：

$$D_{\text{KL}}(P \parallel Q) = \mathbb{E}_{x \sim P} \left[\log \frac{P(x)}{Q(x)} \right] = \mathbb{E}_{x \sim P} [\log P(x) - \log Q(x)]$$

一个和 KL 散度密切联系的量是 **交叉熵** (cross-entropy) $H(P, Q) = H(P) + D_{\text{KL}}(P \parallel Q)$ ，它和 KL 散度很像但是缺少左边一项：

$$H(P, Q) = -\mathbb{E}_{x \sim P} \log Q(x)$$

2.14 结构化概率模型

机器学习的算法经常会涉及到在非常多的随机变量上的概率分布。通常，这些概率分布涉及到的直接相互作用都是介于非常少的变量之间的。使用单个函数来描述整个联合概率分布是非常低效的（无论是计算上还是统计上）。

我们可以把概率分布分解成许多因子的乘积形式，而不是使用单一的函数来表示概率分布。例如，假设我们有三个随机变量 a ， b 和 c ，并且 a 影响 b 的取值， b 影响 c 的取值，但是 a 和 c 在给定 b 时是条件独立的。我们可以把全部三个变量的概率分布重新表示为两个变量的概率分布的连乘形式：

$$p(a, b, c) = p(a)p(b|a)p(c|b)$$

这种分解可以极大地减少用来描述一个分布的参数数量。每个因子使用的参数数目是它的变量数目的指数倍。这意味着，如果我们能够找到一种使每个因子分布具有更少变量的分解方法，我们就能极大地降低表示联合分布的成本。

我们可以用图来描述这种分解。这里我们使用的是图论中的“图”的概念：由一些可以通过边互相连接的顶点的集合构成。当我们用图来表示这种概率分布的分解，我们把它称为结构化概率模型（structured probabilistic model）或者图模型（graphical model）。

有两种主要的结构化概率模型：有向的和无向的。两种图模型都使用图 G ，其中图的每个节点对应着一个随机变量，连接两个随机变量的边意味着概率分布可以表示成这两个随机变量之间的直接作用。

有向（directed）模型使用带有有向边的图，它们用条件概率分布来表示分解，就像上面的例子。特别地，有向模型对于分布中的每一个随机变量 x_i 都包含着一个影响因子，这个组成 x_i 条件概率的影响因子被称为 x_i 的父节点，记为 $Pa_G(x_i)$ ：

$$p(\mathbf{x}) = \prod_i p(x_i | Pa_G(x_i))$$

图 22 给出了一个有向图的例子以及它表示的概率分布的分解。

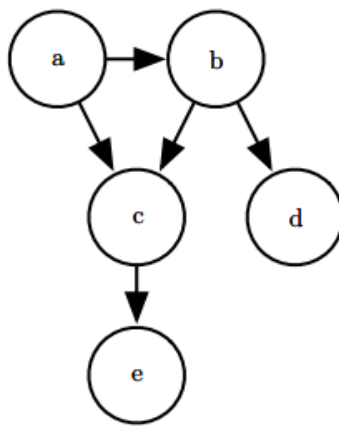


图 22 关于随机变量 a , b , c , d 和 e 的有向图模型。这幅图对应的概率分布可以分解为 $p(a, b, c, d, e) = p(a)p(b|a)p(c|a, b)p(d|b)p(e|c)$ 。该图模型使我们能够快速看出此分布的一些性质。例如， a 和 c 直接相互影响，但 a 和 e 只有通过 c 间接相互影响。

无向（undirected）模型使用带有无向边的图，它们将分解表示成一组函数；不像有向模型那样，这些函数通常不是任何类型的概率分布。 G 中任何满足两两之间有边连接的顶点的集合被称为团。无向模型中的每个团 $C^{(i)}$ 都伴随着一个因子 $\phi^{(i)}(C^{(i)})$ 。这些因子仅仅是函数，并不是概率分布。每个因子的输出都必须是非负的，但是并没有像概率

分布中那样要求因子的和或者积分为 1。

随机变量的联合概率与所有这些因子的乘积成比例（proportional）——意味着因子的值越大则可能性越大。当然，不能保证这种乘积的求和为 1。所以我们需要除以一个归一化常数 Z 来得到归一化的概率分布，归一化常数 Z 被定义为 ϕ 函数乘积的所有状态的求和或积分。概率分布为：

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi^{(i)}(C^{(i)})$$

图 3.8 给出了一个无向图的例子以及它表示的概率分布的分解。

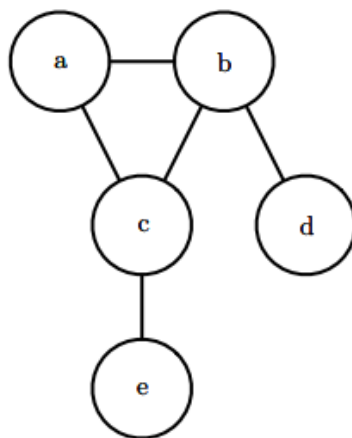


图 23 关于随机变量 a , b , c , d 和 e 的无向图模型。这幅图对应的概率分布可以分解为

$$p(a, b, c, d, e) = \frac{1}{Z} \phi^{(1)}(a, b, c) \phi^{(2)}(b, d) \phi^{(3)}(c, e)$$

。该图模型使我们能够快速看出此分布的一些性质。

例如， a 和 c 直接相互影响，但 a 和 e 只有通过 c 间接相互影响。

2.15 视频主讲内容学习

2.15.1 极大似然估计

（原理：认为抽到的样本就是发生概率最大的）

假设随机变量 $\mathbf{X} \sim P(\mathbf{x}; \theta)$ ，现有样本 x_1, x_2, \dots, x_N ，

定义似然函数为 $\tilde{L} = P(x_1; \theta) P(x_2; \theta) \cdots P(x_N; \theta)$

对数似然函数 $L = \ln \tilde{L} = \ln [P(x_1; \theta) P(x_2; \theta) \cdots P(x_N; \theta)]$

极大似然估计为 $\max L$

例 1: 对于高斯分布 $P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, 有样本 x_1, x_2, \dots, x_N , 使用极大似然估计计算 μ 和 σ^2 。

$$\begin{aligned} L &= \ln \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_1-\mu)^2}{2\sigma^2}} \cdots \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_N-\mu)^2}{2\sigma^2}} \right] \\ &= \ln \left[\left(\frac{1}{\sqrt{2\pi}\sigma} \right)^N e^{-\left(\frac{(x_1-\mu)^2}{2\sigma^2} + \cdots + \frac{(x_N-\mu)^2}{2\sigma^2} \right)} \right] \\ &= -N \ln \sqrt{2\pi} - N \ln \sigma - \left(\frac{(x_1-\mu)^2}{2\sigma^2} + \cdots + \frac{(x_N-\mu)^2}{2\sigma^2} \right) \end{aligned}$$

对 L 求导得

$$\frac{\partial L}{\partial \mu} = 2(x_1 - \mu) + \cdots + 2(x_N - \mu) = 0$$

$$\Rightarrow \mu = \frac{x_1 + x_2 + \cdots + x_N}{N}$$

$$\begin{aligned} \frac{\partial L}{\partial \sigma} &= -\frac{N}{\sigma} + \left(\frac{(x_1-\mu)^2}{\sigma^3} + \cdots + \frac{(x_N-\mu)^2}{\sigma^3} \right) = -\frac{N}{\sigma} + \frac{\sum_{i=1}^N (x_i - \mu)^2}{\sigma^3} = 0 \\ \Rightarrow \sigma^2 &= \frac{\sum_{i=1}^N (x_i - \mu)^2}{N} \end{aligned}$$

2.15.2 误差的高斯分布与最小二乘估计的等价性

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{x}_i \in \mathbb{R}^n$$

$$y_1, y_2, \dots, y_N, y_i \in \mathbb{R}$$

$$y_i = \boldsymbol{\omega}^T \mathbf{x}_i, \boldsymbol{\omega} \in \mathbb{R}^n$$

有拟合误差: $e_i = y_i - \boldsymbol{\omega}^T \mathbf{x}_i$

假设 $e_i \sim N(0,1)$ ，即 $e_i \sim \frac{1}{\sqrt{2\pi}} e^{-\frac{e_i^2}{2}}$

$$\text{似然函数 } L = \ln \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{e_1^2}{2}} \cdots \frac{1}{\sqrt{2\pi}} e^{-\frac{e_N^2}{2}} \right]$$

$$= -N \ln \sqrt{2\pi} - \frac{1}{2} (e_1^2 + e_2^2 + \cdots + e_N^2)$$

最大化 L 等价于最小化 $(e_1^2 + e_2^2 + \cdots + e_N^2)$ ，即

$$J = \min \left((y_1 - \boldsymbol{\omega}^T \mathbf{x}_1)^2 + (y_2 - \boldsymbol{\omega}^T \mathbf{x}_2)^2 + \cdots + (y_N - \boldsymbol{\omega}^T \mathbf{x}_N)^2 \right)$$

求导得

$$\frac{\partial J}{\partial \boldsymbol{\omega}} = (y_1 - \boldsymbol{\omega} \mathbf{x}_1^T) \mathbf{x}_1 + \cdots + (y_N - \boldsymbol{\omega} \mathbf{x}_N^T) \mathbf{x}_N = 0$$

可得

$$\begin{aligned} \boldsymbol{\omega} \left(\sum_{i=1}^N \mathbf{x}_i^T \mathbf{x}_i \right) &= \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i \\ \Rightarrow \boldsymbol{\omega} &= \left(\sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i \right) \left(\sum_{i=1}^N \mathbf{x}_i^T \mathbf{x}_i \right)^{-1} \Leftrightarrow \mathbf{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \end{aligned}$$

还不是
很清楚

第三章 数值计算

机器学习算法通常需要大量的数值计算。这通常是指通过迭代过程更新解的估计值来解决数学问题的算法，而不是通过解析过程推导出公式来提供正确解的方法。常见的操作包括优化（找到最小化或最大化函数值的参数）和线性方程组的求解。对数字计算机来说实数无法在有限内存下精确表示，因此仅仅是计算涉及实数的函数也是困难的。

3.1 上溢和下溢

连续数学在数字计算机上的根本困难是，我们需要通过有限数量的位模式来表示无限多的实数。这意味着我们在计算机中表示实数时，几乎总会引入一些近似误差。在许多情况下，这仅仅是舍入误差。舍入误差会导致一些问题，特别是当许多操作复合时，即使是理论上可行的算法，如果在设计时没有考虑最小化舍入误差的累积，在实践时也可能导致算法失效。

一种极具毁灭性的舍入误差是下溢（underflow）。当接近零的数被四舍五入为零时发生下溢。许多函数在其参数为零而不是一个很小的正数时才会表现出质的不同。例如，我们通常要避免被零除（一些软件环境将在这种情况下抛出异常，有些会返回一个非数字(not-a-number, NaN)的占位符）或避免取零的对数（这通常被视为 $-\infty$ ，进一步的算术运算会使其变成非数字）。

一个极具破坏力的数值错误形式是上溢（overflow）。当大量级的数被近似为 ∞ 或 $-\infty$ 时发生上溢。进一步的运算通常会导致这些无限值变为非数字。

3.2 病态条件

条件数表征函数相对于输入的微小变化而变化的快慢程度。输入被轻微扰动而迅速改变的函数对于科学计算来说可能是有问题的，因为输入中的舍入误差可能导致输出的巨大变化。

考虑函数 $f(\mathbf{x}) = \mathbf{A}^{-1}\mathbf{x}$ ，当 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 具有特征值分解时，其条件数为

$$\max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|$$

这是最大和最小特征值的模之比。当该数很大时，矩阵求逆对输入的误差特别敏感。

这种敏感性是矩阵本身的固有特性，而不是矩阵求逆期间舍入误差的结果。即使我们乘以完全正确的矩阵逆，病态条件的矩阵也会放大预先存在的误差。在实践中，该错误将与求逆过程本身的数值误差进一步复合。

3.3 基于梯度的优化方法

假设我们有一个函数 $y = f(x)$ ，其中 x 和 y 是实数。这个函数的导数（derivative）记为 $f'(x)$ 或 $\frac{dy}{dx}$ 。导数 $f'(x)$ 代表 $f(x)$ 在点 x 处的斜率。换句话说，它表明如何缩放输入的小变化才能在输出获得相应的变化： $f(x + \varepsilon) \approx f(x) + \varepsilon f'(x)$ 。

因此导数对于最小化一个函数很有用，因为它告诉我们如何更改 x 来略微地改善 y 。例如，我们知道对于足够小的 ε 来说， $f(x - \varepsilon \text{sign}(f'(x)))$ 是比 $f(x)$ 小的。因此我们可以将 x 往导数的反方向移动一小步来减小 $f(x)$ 。这种技术被称为梯度下降（gradient descent），如图 24 所示。

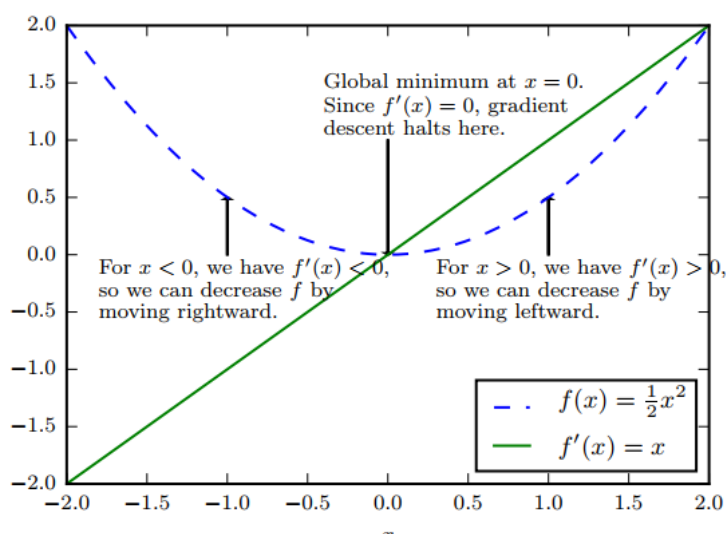


图 24 梯度下降。梯度下降算法如何使用函数导数的示意图，即沿着函数的下坡方向（导数反方向）直到最小。

我们经常最小化具有多维输入的函数： $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 。为了使“最小化”的概念有意义，输出必须是一维的(标量)。

针对具有多维输入的函数，我们需要用到偏导数（partial derivative）的概念。偏导数 $\frac{\partial}{\partial x_i} f(\mathbf{x})$ 衡量点 \mathbf{x} 处只有 x_i 增加时 $f(\mathbf{x})$ 如何变化。梯度（gradient）是相对一个向量求导的导数： f 的导数是包含所有偏导数的向量，记为 $\nabla_{\mathbf{x}} f(\mathbf{x})$ 。梯度的第 i 个元素是 f

关于 x_i 的偏导数。在多维情况下，临界点是梯度中所有元素都为零的点。

在 \mathbf{u} (单位向量) 方向的方向导数 (directional derivative) 是函数 f 在 \mathbf{u} 方向的斜率。换句话说，方向导数是函数 $f(\mathbf{x} + \alpha \mathbf{u})$ 关于 α 的导数 (在 $\alpha = 0$ 时取得)。使用链式法则，我们可以看到当 $\alpha = 0$ 时， $\frac{\partial}{\partial \alpha} f(\mathbf{x} + \alpha \mathbf{u}) = \mathbf{u}^T \nabla_{\mathbf{x}} f(\mathbf{x})$ 。

为了最小化 f ，我们希望找到使 f 下降得最快的方向。计算方向导数：

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{u}^T \mathbf{u} = 1} \mathbf{u}^T \nabla_{\mathbf{x}} f(\mathbf{x}) \\ = \min_{\mathbf{u}, \mathbf{u}^T \mathbf{u} = 1} \|\mathbf{u}\|_2 \|\nabla_{\mathbf{x}} f(\mathbf{x})\|_2 \cos \theta \end{aligned}$$

其中 θ 是 \mathbf{u} 与梯度的夹角。将 $\|\mathbf{u}\|_2 = 1$ 代入，并忽略与 \mathbf{u} 无关的项，就能简化得到

$\min_{\mathbf{u}} \cos \theta$ 。这在 \mathbf{u} 与梯度方向相反时取得最小。换句话说，梯度向量指向上坡，负梯度向量指向下坡。我们在负梯度方向上移动可以减小 f 。这被称为最速下降法 (method of steepest descent) 或梯度下降 (gradient descent)。

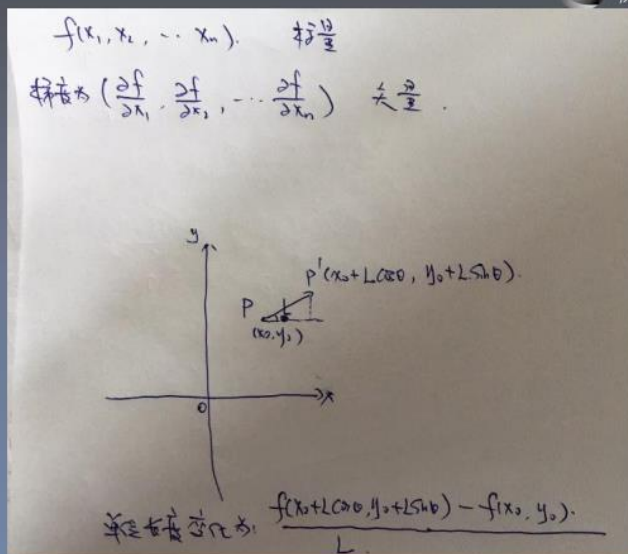
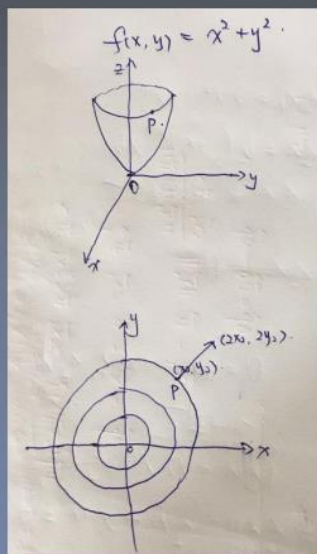
3.4 约束优化

3.4.1 无约束优化

无约束优化是机器学习中最普遍、最简单的优化问题。

$$\mathbf{x}^* = \min_{\mathbf{x}} f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n$$

梯度下降法


 deepshare.net
深度之眼


单位长度变化为: $\frac{f(x_0 + L\cos\theta, y_0 + L\sin\theta) - f(x_0, y_0)}{L}$

$$= \frac{f(x_0 + L\cos\theta, y_0 + L\sin\theta) - f(x_0 + L\cos\theta, y_0)}{L} + \frac{f(x_0 + L\cos\theta, y_0) - f(x_0, y_0)}{L}$$

$$= \sin\theta \cdot \frac{f(x_0 + L\cos\theta, y_0 + L\sin\theta) - f(x_0 + L\cos\theta, y_0)}{L \cdot \sin\theta} + \cos\theta \cdot \frac{f(x_0 + L\cos\theta, y_0) - f(x_0, y_0)}{L \cos\theta}$$

导数定义

$$L \rightarrow 0$$

$$= \sin\theta \cdot \frac{\partial f}{\partial y}(x_0, y_0) + \cos\theta \cdot \frac{\partial f}{\partial x}(x_0, y_0)$$


 deepshare.net
深度之眼

存在上界, 即存在最快

$$(\sin\theta \cdot \frac{\partial f}{\partial y} + \cos\theta \cdot \frac{\partial f}{\partial x})^2 \leq (\sin^2\theta + \cos^2\theta) (\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2)$$

$$\langle x, y \rangle = \|x\| \cdot \|y\| \cos\theta$$

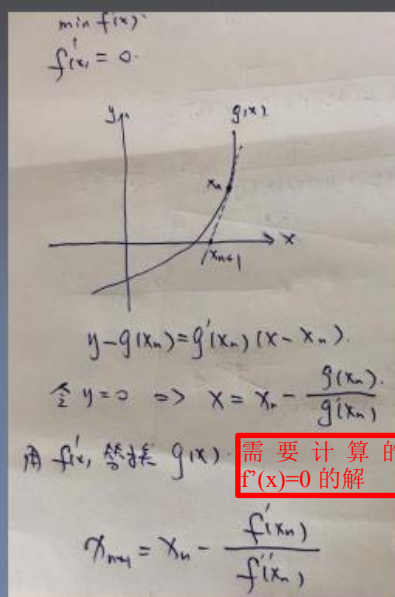
$$\langle x, y \rangle^2 = \|x\|^2 \|y\|^2 \cos^2\theta$$

$$\leq \|x\|^2 \|y\|^2$$

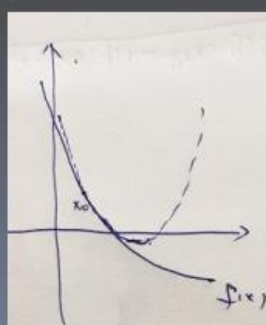
$$\cos\theta \leq \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

条件是 x 和 y 平行

牛顿法（两种解释）



需要计算的是 $f'(x)=0$ 的解



泰勒展开

$$\begin{aligned}
 f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots \\
 &= \frac{f''(x_0)}{2}x^2 + (f'(x_0) - x_0 f''(x_0))x + \dots \\
 -\frac{b}{2a} &= -\frac{f'(x_0) - x_0 f''(x_0)}{f''(x_0)} = x_0 - \frac{f'(x_0)}{f''(x_0)}
 \end{aligned}$$

整理成二次的形式

对称轴位置为最小点

收敛速度比较，梯度下降是一次收敛，牛顿法是二次收敛（速度快，但也有缺陷，要在比较接近最优值的时候才能收敛，否则可能发散）



deepshare.net
深度之眼

$$\begin{aligned}
 \|x_{n+1} - x^*\| &\leq K \|x_n - x^*\| \\
 \|x_{n+1} - x^*\| &\leq K \|x_n - x^*\|^2
 \end{aligned}$$

3.4.2 有约束优化

Karush–Kuhn–Tucker (KKT) 方法是针对约束优化非常通用的解决方案。为介绍 KKT 方法,我们引入一个称为广义 **Lagrangian**(generalized Lagrangian)或广义 **Lagrange** 函数 (generalized Lagrange function) 的新函数。

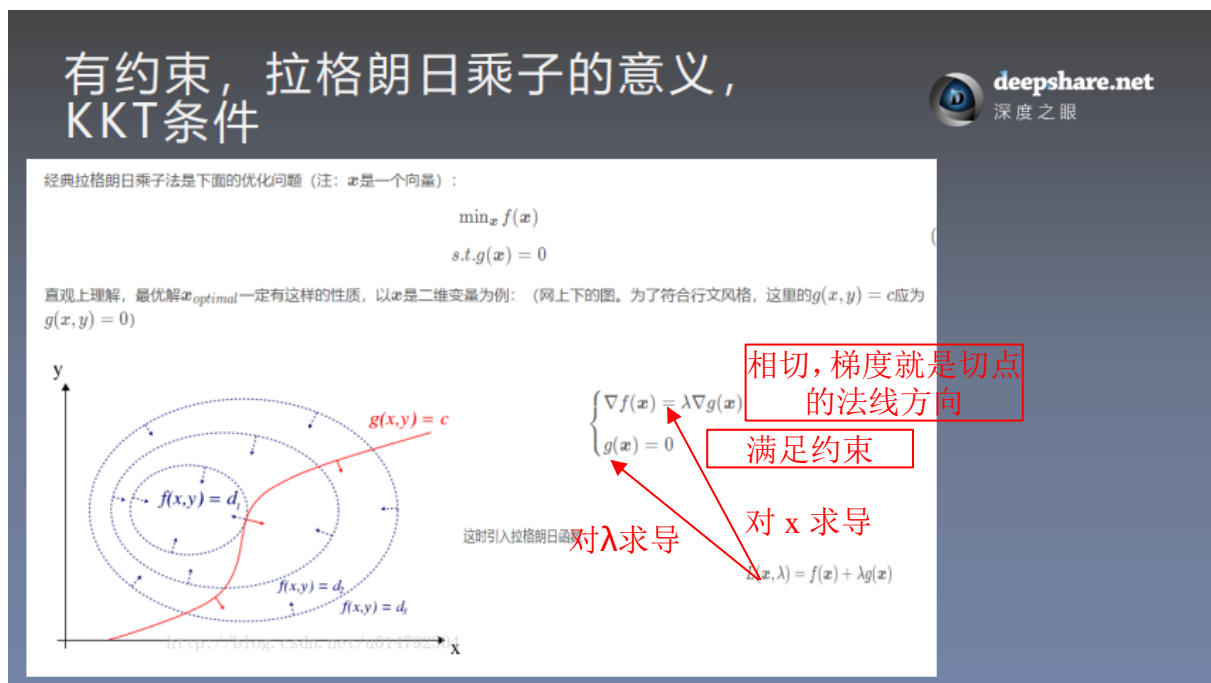
为了定义 Lagrangian，我们先要通过等式和不等式的形式描述 \mathbb{S} 。我们希望通过 m 个函数 $g^{(i)}$ 和 n 个函数 $h^{(j)}$ 描述 \mathbb{S} ，那么 \mathbb{S} 可以表示为 $\mathbb{S} = \{\mathbf{x} \mid \forall i, g^{(i)}(\mathbf{x}) = 0 \text{ and } \forall j, h^{(j)}(\mathbf{x}) \leq 0\}$ 。其中涉及 $g^{(i)}$ 的等式称为等式约束（equality constraint），涉及 $h^{(j)}$ 的不等式称为不等式约束（inequality constraint）。

我们为每个约束引入新的变量 λ_i 和 α_j ，这些新变量被称为 KKT 乘子。广义 Lagrangian 可以如下定义：


$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = f(\mathbf{x}) + \sum_i \lambda_i g^{(i)}(\mathbf{x}) + \sum_j \alpha_j h^{(j)}(\mathbf{x})$$

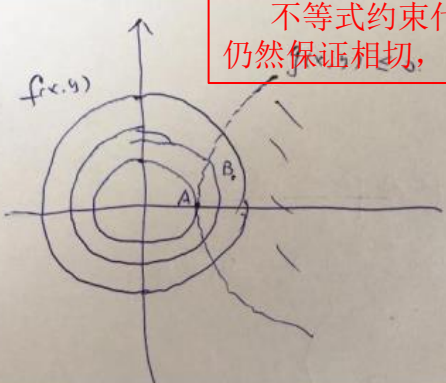
我们可以使用一组简单的性质来描述约束优化问题的最优点。这些性质称为 **Karush–Kuhn–Tucker (KKT) 条件**。这些是确定一个点是最优点的必要条件，但不一定是充分条件。这些条件是：

- 广义 Lagrangian 的梯度为零。
- 所有关于 \mathbf{x} 和 KKT 乘子的约束都满足。
- 不等式约束显示的“互补松弛性”： $\alpha \odot h(\mathbf{x}) = 0$ 。



不等式约束代表的是区域
仍然保证相切，且梯度方向相反


deepshare.net
深度之眼



$$\nabla f(x^*, y^*) = \lambda \nabla g(x^*, y^*)$$

$$\lambda g(x^*, y^*) = 0$$


↓ KKT 条件

$g < 0$ 说明 $\lambda = 0$ ，约束不起作用

$$L = f(x) - \lambda g(x)$$

$$\begin{cases} \frac{\partial L}{\partial x} = 0 \\ \lambda g(x) = 0 \end{cases}$$

N 个约束


deepshare.net
深度之眼

$$\begin{cases} \min f(x) \\ g_i(x) \geq 0, (i=1, 2, \dots, n) \end{cases}$$

$$L = f(x) - \sum_{i=1}^n \lambda_i g_i(x)$$

$$\begin{cases} \frac{\partial L}{\partial x} = 0 & \nabla f(x) = \sum_{i=1}^n \lambda_i \nabla g_i(x) \\ \lambda_i g_i(x) = 0 & (\text{KKT 条件}) \end{cases}$$

λ_i

≥ 0

件基础上，求下列非线性规划问题的 K-T 点：

例 5.1.1 $\min f(x) = 2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2$ ；

s.t. $\begin{cases} x_1^2 + x_2^2 \leq 5, \\ 3x_1 + x_2 \leq 6. \end{cases}$

解 将上述问题的约束条件改写为 $g_i(x) \geq 0$ 的形式：

s.t. $\begin{cases} g_1(x) = -x_1^2 - x_2^2 + 5 \geq 0, \\ g_2(x) = -3x_1 - x_2 + 6 \geq 0. \end{cases}$

设 K-T 点为 $x^* = (x_1, x_2)^T$ ，有


$$\nabla f(x^*) = \begin{bmatrix} 4x_1 + 2x_2 - 10 \\ 2x_1 + 2x_2 - 10 \end{bmatrix},$$

$$\nabla g_1(x^*) = \begin{bmatrix} -2x_1 \\ -2x_2 \end{bmatrix},$$

$$\nabla g_2(x^*) = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.$$

定理 5.1.2，且将(5.1.14)式中第 1 个向量方程拆成分量形式，有：

$$\begin{cases} 4x_1 + 2x_2 - 10 + 2\gamma_1 x_1 + 3\gamma_2 = 0, \\ 2x_1 + 2x_2 - 10 + 2\gamma_1 x_2 + \gamma_2 = 0, \\ \gamma_1(5 - x_1^2 - x_2^2) = 0, \\ \gamma_2(6 - 3x_1 - x_2) = 0, \\ \gamma_1 \geq 0, \\ \gamma_2 \geq 0. \end{cases} \quad (5.1.17)$$



$\begin{cases} x_1 = 1, \\ x_2 = 2, \\ \gamma_1 = 1, \\ \gamma_2 = 0. \end{cases}$

说明约束没有作用，第二个约束 $3*1+2<6$

3.5 线性最小二乘

假设我们希望找到最小化下式的 x 值

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2$$

首先，我们计算梯度：

$$\nabla_x f(x) = A^T(Ax - b) = A^T Ax - A^T b$$

然后，我们可以采用小的步长，并按照这个梯度下降。

算法 4.1 从任意点 x 开始，使用梯度下降关于 x 最小化 $f(x) = \frac{1}{2} \|Ax - b\|_2^2$ 的算法。

将步长 (ϵ) 和容差 (δ) 设为小的正数。

```
while  $\|A^T Ax - A^T b\|_2 > \delta$  do
   $x \leftarrow x - \epsilon (A^T Ax - A^T b)$ 
end while
```

我们也可以使用牛顿法解决这个问题。因为在这个情况下，真实函数是二次的，**牛顿法所用的二次近似是精确的**，该算法会在一步后收敛到全局最小点。

现在假设我们希望最小化同样的函数，但受 $\mathbf{x}^T \mathbf{x} \leq 1$ 的约束。要做到这一点，我们引入 Lagrangian

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda(\mathbf{x}^T \mathbf{x} - 1)$$

现在，我们解决以下问题

$$\min_{\mathbf{x}} \max_{\lambda, \lambda \geq 0} L(\mathbf{x}, \lambda)$$

我们可以用 Moore-Penrose 伪逆： $\mathbf{x} = \mathbf{A}^+ \mathbf{b}$ 找到无约束最小二乘问题的最小范数解。如果这一点是可行，那么这也是约束问题的解。否则，我们必须找到约束是活跃的解。关于 \mathbf{x} 对 Lagrangian 微分，我们得到方程

$$\mathbf{A}^T \mathbf{A} \mathbf{x} - \mathbf{A}^T \mathbf{b} + 2\lambda \mathbf{x} = 0$$

这就告诉我们，该解的形式将会是

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A} + 2\lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{b}$$

λ 的选择必须使结果服从约束。我们可以关于 λ 进行梯度上升找到这个值。为了做到这一点，观察

$$\frac{\partial}{\partial \lambda} L(\mathbf{x}, \lambda) = \mathbf{x}^T \mathbf{x} - 1$$

当 \mathbf{x} 的范数超过 1 时，该导数是正的，所以为了跟随导数上坡并相对 λ 增加 Lagrangian，我们需要增加 λ 。因为 $\mathbf{x}^T \mathbf{x}$ 的惩罚系数增加了，求解关于 \mathbf{x} 的线性方程现在将得到具有较小范数的解。求解线性方程和调整 λ 的过程将一直持续到 \mathbf{x} 具有正确的范数并且关于 λ 的导数是 0。

3.6 作业

一元线性回归的基本假设有哪些？

一、对模型设定的假设

假设 1: 回归模型是正确设定的。

模型的正确设定包括两方面内容：（1）模型选择了正确的变量；（2）模型选择了正确的函数形式。此时，称模型没有设定偏误（specification error）。

二、对解释变量的假设

假设 2: 解释变量 X 是确定性变量, 不是随机变量, 在重复抽样中取固定值。

假设 3: 解释变量 X 在所抽取的样本中具有变异性, 而且随着样本容量的无限增加, 解释变量 X 的样本方差趋于一个非零的有限常数。样本方差的极限为非零的有限常数的假设, 则旨在排除时间序列数据出现持续上升或下降的变量作为解释变量, 因为这类数据不仅使大样本统计推断变得无效, 而且往往产生所谓的伪回归问题(spurious regression problem)。

三、对随机干扰项的假设

假设 4: 随机误差项 μ 具有给定 X 条件下的零均值、同方差不序列相关性, 即

$$\begin{aligned}E(\mu_i | X_i) &= 0 \\ \text{Var}(\mu_i | X_i) &= 0 \\ \text{Cov}(\mu_i, \mu_j | X_i, X_j) &= 0, i \neq j\end{aligned}$$

因此该假设成立时也往往称 X 为外生变量(exogenous explanatory variable), 否则称 X 为内生解释变量(endogenous explanatory variable)。该假设最为重要, 只有该假设成立时, 总体回归函数的随机形式才能等价于非随机形式。

假设 5: 随机误差项与解释变量之间不相关, 即

$$\text{Cov}(X_i, \mu_i) = 0$$

当随机误差项 μ 的条件零均值假设成立时, 该假设一定成立, 因为

$$\text{Cov}(X_i, \mu_i) = E(X_i, \mu_i) - E(X_i)E(\mu_i) = E(X_i, \mu_i) = 0$$

假设 6: 随机误差项服从零均值、同方差的正态分布。

对于随机误差项的正态性假设, 根据中心极限定理, 如果仅包括源生性的随机干扰, 当样本容量趋于无穷大时, 都是满足的。如果包括衍生的随机误差, 即使样本容量趋于无穷大, 正态性假设也经常是不满足的。