目录

Chapter 1

Introduction to data-driven modeling, motivating examples and course outline

（暂定）符号：

- 小写字母：事件实现、列向量（有时）可以使用下划线（例如，$\mathbf{w}$, x, y, z, f, p）。行向量将表示为 $\mathbf{w}^\top$，函数，概率分布。
- 大写字母：随机变量、矩阵，一些函数（cdf）例如 X, Y, Z, F。
- 花体大写字母：集合、算子（例如，$\mathcal{F}$, $\mathcal{L}$, $\mathcal{J}$）

关于期望的说明：$\mathbb{E}[f(x)] := \mathbb{E}_{x\sim p(x)}[f(x)]$。

1 概率基础 Probability Primer

- 频率派（基于长期频率的 UQ）。
- 贝叶斯派（基于概率的纯 UQ）。

概率空间 $(\Omega, \mathcal{F}, p)$，其中 Ω 是样本空间，$\mathcal{F}$ 是来自 σ-代数的事件集合，p 是概率测度 (probability measure)。测度理论是概率理论的理论基础和发展支撑。

2 离散随机变量（取离散值的事件）

$P(A)$ 表示事件 A 为真时的概率。根据定义 $0 \le P(A) \le 1$，即 $P(A) = 0$ 表示该事件绝对不会发生，$P(A) = 1$ 表示该事件绝对会发生。

$$P : \mathcal{F} \to [0, 1] \tag{1.1}$$

$$X : \Omega \to \mathbb{R} \tag{1.2}$$

$P(\overline{A})$ 表示事件非 A 的概率。因此，$P(A) + P(\overline{A}) = 1$。离散随机变量 X 是一个从有限或可数无限离散集 $\mathcal{X}$ 中取值的事件。我们将 $X = x$ 的概率表示为 $P(X = x)$，或简单地表示为 $p(x)$（更正式地表示为 $p(\{w : X(w) = k\})$）。这满足 $0 \le P(A) \le 1$，并且 $\sum_{x\in\mathcal{X}} p(x) = 1$，$p(\{\emptyset\}) = 0$。这里 $p(\cdot)$ 被称为概率质量函数 (probability mass function)。

2.1 概率的基本规则

- Union：$P(A \cup B) = P(A) + P(B) - P(A \cap B)$（或者 $A \cup B$ 意味着 $P(A \cup B) \le P(A) + P(B)$）。
- Joint：$P(A, B) = P(A \cap B) = P(A|B)P(B)$，这也被称为乘法规则，它直接源自条件概率的定义。

- 条件概率：$P(A|B) = \frac{P(A \cap B)}{P(B)}$，其中 $P(B) > 0$。

给定 Joint 分布 $P(A, B)$，我们可以定义边际分布为：

$$P(A) = \sum_{b \in B} p(A, B) = \sum_{b} P(A|B)P(B) = \sum_{b} P(A = a, B = b)P(B = b) \tag{1.3}$$

这被称为求和规则，或全概率规则。

2.2 贝叶斯规则

它是将条件概率的定义与乘法和求和规则结合的结果：

$$P(X = x|Y = y) = \frac{P(X = x \cap Y = y)}{P(Y = y)} \tag{1.4}$$

$$= \frac{P(X = x|Y = y)p(Y = y)}{\sum_{x'} p(X = x'|Y = y)P(Y = y|X = x')} \tag{1.5}$$

2.3 全概率法则

如果 $A_1, ..., A_k$ 是一组互不相交 (disjoint) 的事件（即 $A_i \cap A_j = \emptyset, \forall i \neq j$），且 $\bigcup_{i=1}^{K} A_i = \Omega$，那么 $\sum_{i=1}^{K} P(A_i) = 1$。

2.4 独立性

- X, Y 是（无条件）独立的，如果：$X \perp Y$ 意味着 $P(X, Y) = P(X)P(Y)$。
- X, Y 是条件独立的，如果：$X \perp Y|Z$ 意味着 $P(X, Y|Z) = P(X|Z)P(Y|Z)$。

3 连续随机变量（取连续值的事件）

定义事件 $A = (X \leq a)$, $B(X \leq b)$，以及 $W = (a < X \leq b)$。然后观察到 $B = A \cup W$，$A \cap W = \{0\}$，因此：

$$P(B) = P(A) + P(W) - P(A \cap W) = P(A) + P(W) \tag{1.6}$$

因此

$$P(W) = P(B) - P(A) \tag{1.7}$$

现在我们定义函数 $F(q) := P(X \leq q)$，这被称为累积分布函数 (cumulative distribution function)，或 X 的 cdf。它是一个单调非减 (monotonically non-decreasing) 的函数。然后有 $P(W) = P(a < X \leq b) = F(b) - F(a)$。

假设 cdf 是可微的，我们可以定义 $f(x) = \frac{d}{dx}F(x)$。这被称为概率密度函数，或 X 的 pdf。

根据定义，$P(a < X \leq b) = \int_a^b f(x)dx$。当区间的大小减小时，我们可以写成 $P(x \leq X < x + dx) = P(X)dx$。我们要求 $P(x) > 0$，但可以在给定的 x 的情况下 $P(x) > 1$，只要密度的积分为 1，即：

$$\int_{-\infty}^{\infty} f(x)dx = 1.$$

- $f(x) \geq 0$
- $\int_{-\infty}^{\infty} f(x)dx = 1$
- $\int_{x \in A} f(x)dx = P(x \in A)$

示例：均匀分布，高斯分布。

3.1 分位数

如果 cdf 是单调递增函数，则它具有一个逆函数 F^{-1}。给定 F 是 X 的 cdf，则 $F^{-1}(\alpha)$ 是满足 $P(X \leq x_\alpha) = \alpha$ 的 x_α 的值。这个概率称为 X 的 α-分位数。值 $F^{-1}(0.5)$ 是分布的中位数，左侧和右侧的概率质量各占一半。

我们还可以使用逆 cdf 来计算尾部概率 (tail area probabilities)。

- **示例**：高斯分布。

3.2 均值和方差（量化概率分布或随机变量的属性）

3.2.1 均值，或期望值（1 阶矩）

- 离散随机变量：$\mu = \mathbb{E}[x] := \sum_{x \in X} xP(x)$。
- 连续随机变量：$\mu = \mathbb{E}[x] := \int_x xP(x)dx$。

关于概率测度的：

- $\mathbb{E}[\alpha] = \alpha$ 对于任何常数 $\alpha \in \mathbb{R}$。
- $\mathbb{E}[\alpha f(x)] = \alpha \mathbb{E}[f(x)]$。
- $\mathbb{E}[f(x) + g(x)] = \mathbb{E}[f(x)] + \mathbb{E}[g(x)]$。

3.2.2 方差

衡量分布的离散程度，记作 σ^2（2 阶矩）

$$\sigma^2 = \mathrm{Var}[x] := \mathbb{E}((x-\mu)^2) = \int_x (x-\mu)^2 p(x)dx \tag{1.8}$$

$$= \int_x x^2 p(x)dx + \mu^2 \int_x p(x)dx - 2\mu \int_x xp(x)dx \tag{1.9}$$

$$= \mathbb{E}[x^2] - \mu^2 \tag{1.10}$$

因此：$\mathbb{E}[x^2] = \mu^2 + \sigma^2$。我们还可以定义标准差为：$\mathrm{std}[x] := \sqrt{\mathrm{Var}[x]} = \sqrt{\sigma^2}$。

3.3 一些常见的离散分布

二项分布 $X \sim Bin(n, \theta)$，例如，抛一枚硬币 n 次。如果每次“正面”的概率为 θ，则为二项随机变量。这可以推广到多项分布，适用于非二元结果的情况，例如抛掷一个骰子。

$$\text{pmf: } Bin(k|n,\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k} \text{ 其中} \binom{n}{k} = \frac{n!}{(n-k)!k!}.$$

$$\mathbb{E}[x] = n\theta \text{ 和} \mathrm{Var}[x] = n\theta(1-\theta).$$

伯努利分布 假设我们只抛一次硬币。则 $X \in \{0, 1\}$，如果成功的概率为 θ，则 X 是一个伯努利随机变量：

$$X \sim Ber(\theta) \tag{1.11}$$

$$\text{pmf: } Ber(x|\theta) = \theta^{\mathbb{1}_{x=1}}(1-\theta)^{\mathbb{1}_{x=0}}, \text{ 其中} \mathbb{I}^k \text{ 是指示函数，例如}$$

$$\mathbb{I}_{x=1} = \begin{cases} 1, & x = 1 \\ 0, & \text{其他情况} \end{cases} \tag{1.12}$$

换句话说：

$$Ber(x|\theta) = \begin{cases} \theta, & \text{如果} x = 1 \\ 1-\theta, & \text{如果} x = 0 \end{cases} \tag{1.13}$$

这当然是 $n = 1$ 的二项分布的特例。可以推广到多项分布。

泊松分布 $X \in \{0, 1, 2, ...\}$ 具有参数 $\lambda > 0$ 的泊松分布：$X \sim Poi(\lambda)$。
概率质量函数（pmf）：$Poi(x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$，其中 $e^{-\lambda}$ 作为归一化因子。
这用于建模稀有事件 (rare events) 的计数。（例如，出生、缺陷、突变、车祸等）

3.4 经验分布 / 测度

给定一组数据，$\mathcal{D} = \{x_1, ..., x_N\}$，我们定义经验测度为：

$$P_e(A) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}(A), \tag{1.14}$$

其中 $\delta_x(A)$ 是 Dirac 测度：

$$\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \tag{1.15}$$

我们还可以为每个样本关联权重：

$$P(x) = \sum_{i=1}^{N} w_i\delta_{x_i}(x), \quad 0 \le w_i \le 1, \quad \sum_{i=1}^{N} w_i = 1 \tag{1.16}$$

我们可以将其视为一个直方图，在数据点 x_i 处有“尖峰”，每个尖峰的高度 y 由 w_i 决定。

3.5 一些常见的连续分布

高斯/正态分布

$$X \sim (N)(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{ 其中}\frac{1}{\sqrt{2\pi\sigma^2}} \text{ 作为归一化常数。}$$

并且 $\mu = \mathbb{E}[x]$ 是均值（也是众数），$\sigma^2 = \mathrm{Var}[x]$ 是方差。我们还注意到 μ, σ^2 是充分统计量 (sufficient statistics)。
符号： $X \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow p(X = x) = \mathcal{N}(x|\mu, \sigma^2)$。
特殊情况下，$\mu = 0, \sigma^2 = 1$ 被称为标准正态分布。逆方差通常称为精度：$\lambda = \frac{1}{\sigma^2}$。
cdf 定义为：$\Phi(x; \mu, \sigma^2) = \int_{-\infty}^{x}\mathcal{N}(z|\mu, \sigma^2)dz$，可以用误差函数计算：

$$\phi(x; \mu, \sigma) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \text{ 和}\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt.$$

为什么高斯分布如此流行？

- 计算和解释都很简单（其两个参数是均值和方差!）
- 中心极限定理表明，独立随机变量的和具有近似高斯分布，因此使其成为残差 (residual) 误差/噪声的良好模型。
- 具有非常少的假设（即，具有最大熵）。
- 易于实现。

- Degenerate pdf：$\lim_{\sigma^2 \to 0} \mathcal{N}(x|\mu, \sigma^2) = \delta(x - \mu)$，其中 $\delta(\cdot)$ 是 Dirac delta 函数：

$$\delta_x(x) = \begin{cases} \infty, & \text{如果} x = 0 \\ 0, & \text{如果} x \neq 0 \end{cases} \tag{1.17}$$

并且 $\int_{-\infty}^{\infty} \delta(x)dx = 1$。
平移性质：$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(\mu)$，因为被积函数仅在此处非零。

Student-t 分布

$$T(x|\mu, \sigma^2, \nu) \propto \left(1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}} \tag{1.18}$$

均值 $= \mu$，众数 $= \mu$，方差 $= \frac{\nu\sigma^2}{\nu-2}$ 仅在 $\nu > 2$ 时定义。
重尾，对于小 ν 相较于高斯分布对异常值具有鲁棒性。对于 $\nu = 1$，通常称为 Cauchy 分布。一个常见的选择是 $\nu = 4$。
当 $\nu \gg 5$ 时，它会收敛到高斯分布。

拉普拉斯分布

$$Lap(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right) \tag{1.19}$$

μ 是位置参数，b 是尺度参数。
均值 $= \mu$，众数 $= \mu$，方差 $= 2b^2$。
重尾，鲁棒性，相较于高斯分布/Student-t 分布，在零附近集中更多质量。
最后一个属性在希望在模型中鼓励稀疏性的情况下非常有用！
更多情况：Gamma，Beta，卡方，Pareto。

4 协方差和相关性

* 联合分布 $p(x_1, ..., x_D)$
两个随机变量 X 和 Y 之间的协方差测量 X 和 Y 的线性关系程度：

$$cov[x, y] = \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y] \tag{1.20}$$

如果 $\mathbf{x}$ 是一个 d 维随机向量，则其协方差矩阵是一个对称的、正定的矩阵，定义为：

$$cov[\mathbf{x}] = cov[\mathbf{x}, \mathbf{x}] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top] = \begin{bmatrix} \text{var}[x_1] & cov[x_1, x_2] & \cdots & cov[x_1, x_d] \\ cov[x_2, x_1] & \text{var}[x_2] & \cdots & cov[x_2, x_d] \\ \vdots & \vdots & \ddots & \vdots \\ cov[x_d, x_1] & cov[x_d, x_2] & \cdots & \text{var}[x_d] \end{bmatrix} = \Sigma = \Lambda^{-1} \tag{1.21}$$

其中 Σ 是协方差矩阵，Λ 是精度矩阵。协方差的值可以在零和无穷大之间取。
有时使用具有有限上界的归一化测度更为可取。

4.1 皮尔逊相关系数

$$corr[x, y] = \frac{cov[x, y]}{\sqrt{var[x]var[y]}} \quad -1 \leq corr[x, y] \leq 1. \tag{1.22}$$

具体而言，可以证明当且仅当 $Y = aX + b$ 对于某些 a, b 时，有 $corr[x, y] = 1$。如果 X 和 Y 独立，则 $corr[x, y] = 0$。

4.2 多元高斯分布

最广泛使用的连续随机变量的联合概率密度函数。
在 D 维中的 pdf:

$$\mathbf{x} = (x_1, ..., x_D) \sim \mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{1.23}$$

其中 $\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}$ 作为归一化常数，$\mu = \mathbb{E}[\mathbf{x}] \in \mathbb{R}^D$，$\Sigma = cov[\mathbf{x}, \mathbf{x}]$ 是 $D \times D$ 的协方差矩阵。$\Lambda = \Sigma^{-1}$ 是精度矩阵 (precision matrix)。

5 随机变量的变换

$$x \sim p(x), y = f(x), p(y)?$$

5.0.1 线性变换

$$y = f(x) = Ax + b \Rightarrow \mathbb{E}[y] = \mathbb{E}[Ax + b] = A\mu + b, \tag{1.24}$$

其中 $\mu = \mathbb{E}[x]$ 由于期望的线性特性。
如果 $y = a^\top x + b$，则 $\mathbb{E}[y] = a^\top \mu + b$。同样，$cov[y] = cov[Ax + b] = A^\top \Sigma A$，其中 $\Sigma = cov[x]$。
如果 $f(\cdot)$ 是标量值，这变为：$cov[y] = var[a^\top x + b] = a^\top \Sigma a$。

5.0.2 一般变换

- 离散随机变量：如果 X 是离散随机变量，则我们可以推导 y 的 pmf。$p_y(y) = \sum_{x:f(x)=y} p_x(x)$
- 连续随机变量：我们无法使用上述方法，因为我们在对密度进行求和。相反，我们使用 cdf。

$$F_y(y) = p(Y \le y) = p(f(x) \le y) = p(x \in \{x|f(x) \le y\}) \tag{1.25}$$

然后我们可以通过对 cdf 求导获得 pdf（y 的）：

$$F_y(y) = p(Y \le y) = p(x \le f^{-1}(y)) = F_x(f^{-1}(y)) \tag{1.26}$$

对其进行求导得到：

$$p_y(y) = \frac{d}{dy}F_y(y) = \frac{d}{dy}F_x(f^{-1}(y)) = \frac{dx}{dy}\frac{d}{dx}F_x(x) = \frac{dx}{dy}p_x(x), \tag{1.27}$$

其中 $x = f^{-1}(y)$。一般情况下：$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right|$。
这可以扩展到多元情况：$p_y(y) = p_x(x)\left|\det \frac{\partial x}{\partial y}\right|$，即逆映射的雅可比 (Jacobian matrix)。

5.1 中心极限定理

考虑 N 个随机变量，其 pdf 为 $p(x_i)$（不一定是高斯的!），每个随机变量的均值为 μ，方差为 σ^2。我们假设每个变量是独立同分布的，即 iid。设 $S_N = \sum_{i=1}^N x_i$ 为随机变量的和。那么，

$$p(S_N = s) = \frac{1}{\sqrt{2\pi N\sigma^2}} \exp\left(-\frac{(s - N\mu)^2}{2N\sigma^2}\right) \quad 当N \to \infty. \tag{1.28}$$

因此，$Z_N := \frac{S_N - N\mu}{\sigma\sqrt{N}} = \frac{\bar{x}-\mu}{\sigma/\sqrt{N}}$ 的分布收敛到标准正态分布。这里 $\bar{x} := \frac{1}{N}\sum_{i=1}^N x_i$ 是样本均值。

6 蒙特卡洛近似

一般来说，使用变换公式计算随机变量函数的分布可能是困难的。一种强有力的替代方法是采样。

我们首先从 $p_x(\cdot)$ 生成 S 个样本 $x_1, ..., x_s$（可以是 iid 或者相关的，例如 mcmc）。然后，我们可以使用 $\{f(x_s)\}_{s=1}^S$ 的经验分布来近似 $p_y(y)$。

使用蒙特卡洛近似计算期望

$$\mathbb{E}_{x\sim p(x)}[f(x)] = \mathbb{E}[f(x)] = \int f(x)p(x)dx \approx \frac{1}{S}\sum_{s=1}^{S} f(x_s) \tag{1.29}$$

通过改变函数 $f(\cdot)$，我们可以计算不同的感兴趣的量：

- $\bar{x} = \frac{1}{S}\sum_{s=1}^S x_s \approx \mathbb{E}[x]$。
- $\frac{1}{S}\sum_{s=1}^S (x_s - \bar{x})^2 \approx Var[x]$。
- $\frac{1}{S}\mathbb{I}\{x_s \le c\} \to p(x \le c)$，中位数 $\{x_1, ..., x_s\} \to median(x)$。
- 熵和相对熵：$x \sim p(\cdot)$ $y \sim q(\cdot)$
  $\mathcal{H}[X] = -\int p(x)\log p(x)dx$，$\mathcal{H}[X|Y] = -\int p(x)\log\frac{p(x)}{q(x)}dx = KL[p||q]$ 是 Kullback-Leibler 散度。
- 互信息：$I(X,Y) = \int_x\int_y p(x,y)\log\frac{p(x,y)}{p_x(x)p_y(y)}dxdy$ （$=0$ 如果 X 和 Y 是独立的）。

6.1 最大似然估计 (MLE)

估计分布/模型参数的重要技术。

Setup 给定一些数据 $\mathcal{D} := \{x_1, ..., x_N\}$，$x_i \in \mathbb{R}^d$。
假设一个分布族 $p_\theta(x), \theta \in \Theta$。
假设数据的概率模型，即：

$$x \overset{iid}{\sim} p_\theta(x)$$

对于某个 θ。

目标 Goal 估计最能解释观察到的数据的真实值 θ。

定义 如果 $\theta_{MLE} = \arg\max_{\theta\in\Theta} p(\mathcal{D}|\theta)$，则 θ_{MLE} 是最大似然估计，其中 $p(\mathcal{D}|\theta)$ 是似然函数。更准确地说，$p(\mathcal{D}|\theta_{MLE}) = \max_{\theta\in\Theta} p(\mathcal{D}|\theta)$。

$$p(\mathcal{D}|\theta) = p(x_1, ..., x_N|\theta) = \prod_{i=1}^N p(x_i|\theta) = \prod_{i=1}^N p(X = x_i|\theta)$$

备注 Remark

1. 一个 MLE 可能不是唯一的。
2. MLE 可能不存在（最大值可能不会在某个 $\theta \in \Theta$ 中实现）。

优点

- 通常计算简单且往往易于解释（例如，随机变量的均值是样本均值）。
- 良好的渐近性质：
  - **一致性**：当 $N \to \infty$ 时，收敛到真实的 θ 的概率。

- **渐近正态性**：当 $N \to \infty$ 时，其分布收敛于正态分布。
- **有效性**：即它具有最低的渐近方差。
- **在重新参数化下的不变性**：即对于 $\forall g : g(\theta_{MLE})$ 是 $g(\theta)$ 的 MLE。

缺点

- 这是一个点估计（没有不确定性的表示）。理想情况下，我们希望计算关于 θ 的后验：$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$（贝叶斯）。
- 可能会遇到奇怪的情况（缺乏鲁棒性）。
- 过拟合。
- 存在性和唯一性没有保证。

6.2 针对一维高斯的 MLE

Setup 给定 $\mathcal{D} := \{x_1, ..., x_N\}, x_i \in \mathbb{R}$。

假设

$$x \overset{iid}{\sim} p_\theta(x) = \mathcal{N}(\mu, \sigma^2), \quad \theta := \{\mu, \sigma^2\}$$

似然 Likelihood

$$p(\mathcal{D}|\theta) = p(x_1, ..., x_N|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

$$= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2\right) \tag{1.1}$$

$$\theta_{MLE} = \arg\max p(\mathcal{D}|\theta) = \arg\max \log p(\mathcal{D}|\theta)$$

$$= -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{1.2}$$

取梯度并设为零：

$$0 = \frac{\partial}{\partial\mu}\log p(\mathcal{D}|\theta) \Rightarrow \frac{1}{\sigma^2}\sum_{i=1}^{N} 2(x_i - \mu) = 0 \Rightarrow \sum_{i=1}^{N} x_i - N\mu = 0$$

$$\Rightarrow \mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1.3}$$

这个临界点是最大值吗?

$$\frac{\partial^2}{\partial\mu^2}\log p(\mathcal{D}|\theta) = -\frac{N}{\sigma^2} < 0,$$

因此 μ_{MLE} 是全局唯一的极大值点。

Chapter 2

概率论基础通过贝叶斯线性回归的视角

1 统计比较

为了比较两个数字，我们可以观察它们的差异或比率。如果我们想比较两个概率密度，也可以使用密度差异或密度比率。我们在这里的重点将放在密度比率上，因为它们在机器学习中无处不在。

$$r(x) := \frac{p(x)}{q(x)} \tag{2.1}$$

是两个概率密度 p 和 q 之间的密度比率。从直观上讲，这告诉我们“我们需要多少”来修正 $q(x)$ 使其匹配 $p(x)$，即 $p(x) = r(x)q(x)$，其中 $r(x)$ 是修正因子。

这在很多情况下都会出现！例如：

- (1) 贝叶斯定理：$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)p(y)}{p(x)}$。
- (2) 散度和最大似然：

$$KL[p(x)||q(x)] = \int p(x) \log \frac{p(x)}{q(x)} dx \rightarrow \text{最大似然估计等同于最小化此散度。}$$

- (3) 重要性采样：

$$\int p(y|x)p(x)dx = \int p(y|x)q(x)\frac{p(x)}{q(x)}dx = \mathbb{E}_{x\sim q(x)}[p(y|x)r(x)].$$

- (4) 互信息：

$$I(X,Y) = \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy = KL[p(x,y)||p(x)p(y)]$$

其中 $p(y) = \int p(y|x)p(x)dx$，所以我们有

$$I(X,Y) = \int p(y|x)p(x) \log \frac{p(y|x)}{p(y)} dxdy = \mathbb{E}_{y\sim p(y)} \left[KL[p(x|y)||p(x)] \right].$$

2 线性回归

- 它是统计学的” 工作马 (Workhorse)”。
- 这不仅仅是关于直线和平面！

设置 Setup： 给定 $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$，$x_i \in \mathbb{R}^d$，$y_i \in \mathbb{R}$。
目标 Goal： 选择 $f : \mathbb{R}^d \rightarrow \mathbb{R}$ 来预测新 x 的 y 值。

备注 Remark： 使用基函数 (basis function)，我们可以表示非线性函数。例如，

$$f(x) = w^\top x, \quad w \in \mathbb{R}^d \Rightarrow f(x) = \sum_{j=1}^{d} w_j x_j; \quad \text{或} \quad z = \varphi(x), \quad z : \mathbb{R}^d \to \mathbb{R}^m = f(x) = \sum_{j=1}^{m} w_j \varphi_j(x)$$

其中 $\varphi(x) = (\varphi_1(x), ..., \varphi_m(x))$ 是特征或基函数，例如多项式、傅里叶、波浪、径向基函数等。

线性意味着模型在参数 w 中是线性的!

是时候做出一些选择并定义我们的模型： 我们选择一种判别模型（与生成模型相比）来建模 $p(y|x)$。首先假设一个由 $\theta \in \Theta$ 参数化的族 $p(y|x)$，并使用 $\mathcal{D}$ 来估计 θ。

使用哪个族? 我们第一次选择将是高斯分布! 因此：$p_\theta(y|x) = \mathcal{N}(y|\mu(x), \sigma^2(x)), \quad \theta = (\mu, \sigma^2)$（函数在 $\mathbb{R}^d$ 上）。请注意，这种表示法只是为了方便而采用的简写。

我们应该使用什么样的 $\mu(x)$ 和 $\sigma^2(x)$?

高斯线性回归模型通过假设以下关系来建模数据 $\mathcal{D}$:

$$p(y|x; w) = \mathcal{N}(y|w^\top x, \sigma^2), \qquad \mu(x) = w^\top x,\ \sigma^2(x) = \sigma^2,$$
$$\theta = \{w, \sigma^2\},\ w \in \mathbb{R}^d,\ \sigma^2 > 0$$

另外，可以将此模型视为:

$$y = w^\top x + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2),$$

即，噪声被建模为高斯随机变量。

此外，还可以做出其他选择，例如拉普拉斯分布用于鲁棒回归。

使用最大似然估计（MLE）估计 θ

设置 Setup： 给定 $\mathcal{D} = ((x_1, y_1), ..., (x_n, y_n))$，$x_i \in \mathbb{R}^d$，$y_i \in \mathbb{R}$，$\theta = (\mu, \sigma^2)$（我们将假设 σ^2 是已知的）。

模型 Model： 我们假设 $y_1, ..., y_n$ 是随机独立同分布变量，其模型为

$$y_i \sim \mathcal{N}(w^\top x_i, \sigma^2).$$

那么 $\theta_{MLE} = \arg\max_{\theta \in \Theta} p(\mathcal{D}|\theta)$，其中 $p(\mathcal{D}|\theta)$ 是似然函数。现在我们需要根据模型写下我们的似然函数。（注意，只有 y 是随机的，X 是确定的。）

$$\begin{aligned} p(\mathcal{D}|\theta) &= p(y_1, ..., y_n | x_1, ..., x_n, \theta) \\ &= \prod_{i=1}^{n} p(y_i|x_i, \theta) = \prod_{i=1}^{n} \mathcal{N}(y_i|w^\top x_i, \sigma^2) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - w^\top x_i)^2\right) \\ \Rightarrow p(\mathcal{D}|\theta) &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - w^\top x_i)^2\right) \end{aligned}$$

在这里，我们应该注意到我们可以将 $\sum_{i=1}^{n}(y_i - w^\top x_i)^2$ 写为 $(y - Xw)^\top(y - Xw) = \|y - Xw\|$，其中 X 是 $(n \times d)$ 的设计矩阵。

$$\Rightarrow -\log p(\mathcal{D}|\theta) = \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(y - Xw)^\top(y - Xw) = \mathcal{L}(w)$$

$$= \frac{1}{2}y^\top y - (Xw)^\top y + \frac{1}{2}w^\top X^\top X w.$$

因此：

$$\nabla_w \mathcal{L}(w) = -X^\top y + X^\top X w = -X^\top y + X^\top X w.$$

设置为零 yield：$\nabla_w \mathcal{L}(w) = 0 \Rightarrow w_{MLE} = (X^\top X)^{-1} X^\top y$, 其中 $(X^\top X)$ 需要是可逆的。下面的讨论：$X^\top X$ 是可逆的，当 X 的列线性无关时。

$(X^\top X)^{-1} X^\top := X^\dagger$ 被称为摩尔-彭若斯伪逆。要检查这是否确实是一个最小值，我们可以检查海森矩阵：

$H = \nabla_w^2 \mathcal{L}(w) = X^\top X \rightarrow$ 这是对称正半定 (PSD) 的。假设$X^\top X$ 是可逆的，则$\mathcal{L}$ 在w 中是严格凸的，w_{MLE} 是唯一的 minin

几何解释为在特征空间上的投影。

基函数 MLE:

$$w_{MLE} = (\Phi^\top \Phi)^{-1} \Phi^\top y, \quad \varphi : \mathbb{R}^d \rightarrow \mathbb{R}.$$

其中

$$\phi = [\varphi_1(x), ..., \varphi_m(x)] \in \mathbb{R}^{(n \times m)}.$$

3 贝叶斯线性回归

3.1 为什么选择贝叶斯线性回归？

- MLE 容易过拟合!
- 在 w 上放置一个先验并使用最大后验估计 (maximum a posteriori probability, MAP) 会解决过拟合问题，但我们也希望有一些不确定性的表示，即能够访问预测分布 $p(y^*|x^*, \mathcal{D})$。

设置 Setup： 给定 $\mathcal{D} = ((x_1, y_1), ..., (x_n, y_n)), x_i \in \mathbb{R}^d, y_i \in \mathbb{R}, \theta = (\mu, \sigma^2)$（我们将假设 σ^2 是已知的）。
模型 Model： $y_1, ..., y_n$ 在给定 w 时是 iid 的，且 $y_i \sim \mathcal{N}(w^\top x_i, a^{-1})$，$a = \frac{1}{\sigma^2}$ 被称为精度。对于 $w \sim \mathcal{N}(0, b^{-1} I)$，在 $w \in \mathbb{R}^d$ 上的多变量（独立因为协方差是对角的）高斯先验。
我们还将假设 a, b 是已知的，因此模型参数仅为权重 $\theta = \{w\}$。
备注 Remark： 我们也可以使用基函数来建模非线性。

3.2 似然函数：

$$p(\mathcal{D}|\theta) \propto \exp\left(-\frac{a}{2}(y - Xw)^\top (y - Xw)\right)$$

其中 $y = (y_1, ..., y_n)$ 且

$$X = \begin{bmatrix} x_{11} & ... & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & ... & x_{nd} \end{bmatrix}.$$

3.3 后验分布（根据贝叶斯定理）：

$$p(\mathcal{D}|\theta) \propto p(\mathcal{D}|w) p(w)$$

$$p(\mathcal{D}|\theta) \propto \exp\left(-\frac{a}{2}(y - Xw)^\top (y - Xw) - \frac{b}{2} w^\top w\right).$$

注意到指数中是关于 w 的二次项。这表明后验是高斯分布。我们实际上可以通过“完成平方”来推导这个结果。为了看到这一点，让我们重新写：

$$a(y-Xw)^\top(y-Xw)-bw^\top w = a(y^\top y-2w^\top X^\top y+w^\top X^\top Xw)+bw^\top w$$
$$= ay^\top y-2aw^\top X^\top y+w^\top(aX^\top X+bI)w.$$

我们可以通过注意到以下事实来使其看起来像高斯的指数：一般而言，$(x-\mu)^\top\Lambda(x-\mu)=x^\top\Lambda x-2x^\top\Lambda\mu+\mu^\top\Lambda\mu$ 其中最后一项是常数。
为了匹配项，我们设定 $\Lambda := aX^\top X+bI$（精度矩阵）。

我们还希望满足 $aw^\top X^\top y = w^\top\Lambda\mu$ 或 $aX^\top y=\Lambda\mu$，因此令 $\mu := a\Lambda^{-1}X^\top y$。因此，后验是高斯分布！

$$p(w|\mathcal{D})=\mathcal{N}(w|\mu,\Lambda^{-1}),$$

其中

$$\Lambda = aX^\top X+bI,\quad \mu = a\Lambda^{-1}X^\top y.$$

MAP 估计：$w_{MAP}=\arg\max p(w|\mathcal{D})$，由于 $p(w|\mathcal{D})$ 是高斯分布，因此 MAP 估计是其众数，例如，

$$w_{MAP}=\mu=a(aX^\top X+bI)^{-1}X^\top y=(X^\top X+\frac{b}{a}I)^{-1}X^\top y.$$

将其与 w_{MLE} 进行比较，以查看正则化的效果！这里的 $\frac{b}{a}$ 项作为正则化项。还要注意，最大化 $p(w|\mathcal{D})$ 等价于最小化 $\|y-Xw\|_2^2+\lambda\|w\|_2^2$，其中 $\lambda=\frac{b}{a}$。

4 贝叶斯线性回归的预测分布

给定一个新的测试点 x^*，我们希望预测对应的 $y^*: p(y^*|x^*,\mathcal{D})$。回想一下，根据我们的构造，$y^*\sim\mathcal{N}(w^\top x^*,a^{-1})$ 与 $y_1,...,y_n$ 独立。一般来说，预测分布是通过边缘化掉任何不确定变量或参数来计算的。在这里，这转换为：

$$p(y^*|x^*,\mathcal{D})=\int p(y^*|x^*,\mathcal{D},w)p(w|x^*,\mathcal{D})dw$$
$$=\int\mathcal{N}(y^*|w^\top x^*,a^{-1})\mathcal{N}(w|\mu,\Lambda^{-1})dw$$
$$\propto\int\exp\left(-\frac{a}{2}(y-Xw)^\top(y-Xw)\right)\exp\left(-\frac{1}{2}(w-\mu)^\top\Lambda(w-\mu)\right)dw.$$

注意，条件在 x 上只是为了增强清晰的符号。统计上，y 与 x 没有依赖关系，因为 x 是确定的。

我们的目标是将这个积分转换成形式 $\int\mathcal{N}(w|...)\mathcal{N}(y^*)dw=g(y^*)\propto\mathcal{N}(y^*|...)$。为此，我们将采用“完成平方”的常用技巧。经过一些繁琐的推导，我们可以得到最终结果：

$$p(y^*|x^*,\mathcal{D})=\mathcal{N}(y^*|u,\frac{1}{\lambda}),$$

其中

$$u=\mu^\top x^*,\quad \mu=w_{MAP}=(X^\top X+\Lambda)^{-1}X^\top y,$$
$$\frac{1}{\lambda}=\frac{1}{a}+x^\top\Lambda^{-1}x,\quad \Lambda=aX^\top X+bI.$$

Chapter 3

优化：梯度与 Hessian，梯度下降，牛顿算法，随机梯度下降

1 优化

设置 Setup：给定一个模型，参数为 $\theta = (\theta_1, \theta_2, ..., \theta_l)$ 和损失函数 $J(\theta)$，我们的目标是确定 θ^* 使得：

$$\theta^* = \arg\min J(\theta)$$

我们需要确定临界点，即 $\nabla_\theta J(\theta) = 0$。这个条件在最小值、最大值和鞍点情况下都成立。

1.1 梯度

$$\nabla_\theta f(\theta) = \begin{bmatrix} \frac{\partial f(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial f(\theta)}{\partial \theta_d} \end{bmatrix}$$

梯度下降：

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta J(\theta_n)$$

其中 η 是步长（通常称为学习率）。$\theta_{n+1} \in \mathbb{R}^{d\times 1}, \theta_n \in \mathbb{R}^{d\times 1}$ 和 $\nabla_\theta J(\theta_n) \in \mathbb{R}^{d\times 1}$。这是一个一阶方法，因为它依赖于围绕 θ 的 $J(\theta)$ 的线性近似。也许对 $J(\theta)$ 进行高阶近似会导致更快的收敛？是的！

1.2 Hessian 矩阵

$$\nabla^2_\theta f(\theta) = \begin{bmatrix} \frac{\partial^2 J(\theta)}{\partial \theta_1^2} & \frac{\partial^2 J(\theta)}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 J(\theta)}{\partial \theta_1 \partial \theta_d} \\ \frac{\partial^2 J(\theta)}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 J(\theta)}{\partial \theta_2^2} & \cdots & \frac{\partial^2 J(\theta)}{\partial \theta_2 \partial \theta_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J(\theta)}{\partial \theta_d \partial \theta_1} & \frac{\partial^2 J(\theta)}{\partial \theta_d \partial \theta_2} & \cdots & \frac{\partial^2 J(\theta)}{\partial \theta_d^2} \end{bmatrix}$$

是一个 $(d \times d)$ 矩阵。

where $g_n := \nabla_\theta J(\theta_n)$ and $H_n := \nabla^2_\theta J(\theta_n)$. 让我们使用 θ_n 附近的 $J(\theta)$ 的泰勒展开：

$$\begin{aligned} J(\theta) &\approx J(\theta_n) + g_n^\top(\theta - \theta_n) + \frac{1}{2}(\theta - \theta_n)^\top H_n(\theta - \theta_n) \\ &= J(\theta_n) + g_n^\top(\theta - \theta_n) + \frac{1}{2}\left[\theta^\top H_n \theta - 2\theta^\top H_n \theta_n + \theta_n^\top H_n \theta_n\right] \end{aligned}$$

最小值应该满足 $\nabla_\theta J(\theta) = 0$，因此：

$$-g_n^\top = H_n\theta - H_n\theta_n \Rightarrow \theta - \theta_n = -H_n^{-1}g_n.$$

牛顿算法：

$$\theta_{n+1} = \theta_n - H_n^{-1}g_n$$

通过使用曲率信息，比梯度下降更好地利用了几何特性。

例子：线性回归 $p(y|x^\top x, \sigma^2)$

回顾线性回归的损失函数：

$$\mathcal{L}(w) = \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(y - Xw)^\top(y - Xw).$$

然后

$$\nabla_w\mathcal{L}(w) = -X^\top y + X^\top Xw$$

和

$$\nabla_w^2\mathcal{L}(w) = X^\top X.$$

因此，参数更新为：

$$\text{梯度下降：} \quad w_{n+1} = w_n - \eta[X^\top Xw_n - X^\top y] \tag{3.13}$$

$$\text{牛顿：} \quad w_{n+1} = w_n - (X^\top X)^{-1}[X^\top Xw_n - X^\top y]. \tag{3.14}$$

备注 Remark： 对于 σ^2 也可以进行类似的更新！η 对于每个参数可以不同（也可以在迭代过程中自适应变化！）也可以通过“线搜索 (line search) ”方法进行严格调优，但在机器学习应用中并不实用。

局限性：

- 梯度下降收敛缓慢。选择 η 是一门艺术。
- 大数据的可扩展性。
- 精确的 Hessian 通常很难计算 $\rightarrow$ 准牛顿方法。
- 在病态曲率 (pathological curvature) 情况下表现不佳。

2 随机梯度下降

在许多机器学习应用中，损失函数在数据点上进行因式分解，即可以写成：

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} J_i(\theta) \tag{3.15}$$

（即，见线性回归）。

每一项 $J_i(\theta)$ 通常与第 i 个观察值/数据点相关。在这种情况下，标准的“全批次”梯度下降方法将采取以下形式：

$$\theta_{n+1} = \theta_n - \eta\nabla_\theta J(\theta_n) = \theta_n - \eta\frac{1}{n}\sum_{i=1}^{n}\nabla_\theta J^{(i)}(\theta_n) \tag{3.16}$$

在随机梯度下降中，真实梯度通过单个样本的梯度进行近似：

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta J^{(i)}(\theta_n) \tag{3.17}$$

在两者之间的折中是对一组“迷你批次”数据的梯度进行近似：

$$\theta_{n+1} = \theta_n - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta J(\theta_n) \tag{3.18}$$

对整个数据集进行一次完整的循环称为一个“周期”。

SGD 的收敛性已使用凸优化和随机近似理论进行了分析，已知在适当的学习率和一些温和的假设下，对于凸问题它收敛到全局最小值，否则 (non-convex) 收敛到局部最小值。

2.1 梯度下降变体

2.1.1 带动量的 SGD

$$u_{n+1} = \gamma u_n + \eta \nabla_\theta J(\theta_n) \tag{3.19}$$

$$\theta_{n+1} = \theta_n - u_{n+1} \tag{3.20}$$

其中：

- η 是学习率。
- $\gamma = 0 \rightarrow$ 标准梯度下降。
- $\gamma = 0.9$ 是应用中常用的典型值。

直觉：平滑陡峭谷底的振荡。

2.1.2 Nesterov 加速

$$u_{n+1} = \gamma u_n + \eta \nabla_\theta J(\theta_n - \gamma u_n) \tag{3.21}$$

$$\theta_{n+1} = \theta_n - u_{n+1} \tag{3.22}$$

直觉：防止动量过快。

2.2 自适应学习率方法

2.2.1 RMSprop

$$\mathbb{E}[g^2]_n := \text{迭代} n \text{ 的平方梯度的平均值}, \quad g_n := \nabla_\theta J(\theta_n).$$

$$\mathbb{E}[g^2]_{n+1} = \gamma \mathbb{E}[g^2]_n + (1-\gamma) g_n^2, \tag{3.23}$$

$$\theta_{n+1} = \theta_n - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_n + \epsilon}} g_n. \tag{3.24}$$

这是标准梯度下降更新，具有自适应学习率。通常 $\gamma = 0.9$ 和 $\eta = 0.001$。

2.2.2 Adam

$$m_{n+1} = \theta_1 m_n + (1-\theta_1) g_n, \tag{3.25}$$

$$v_{n+1} = \theta_2 v_n + (1-\theta_2) g_n^2. \tag{3.26}$$

对梯度的一阶矩（均值）和二阶矩（方差）的估计。这里 m_0 和 v_0 通常初始化为零，上述估计会偏向于零。为了抵消这种偏差，我们可以考虑修正：

$$\hat{m}_n = \frac{m_n}{1-\theta_1^n}, \tag{3.27}$$

$$\hat{v}_n = \frac{v_n}{1-\theta_2^n}, \tag{3.28}$$

并执行更新：

$$\theta_{n+1} = \theta_n - \frac{\eta}{\sqrt{\hat{v}_n + \epsilon}} \hat{m}_n. \tag{3.29}$$

3 逻辑回归分类（在生物统计学、医学、社会科学中的许多应用）

示例：假设你是一名精算师 (actuary)，你想构建一个模型来预测某人未来 10 年内可能去世的概率：

$$p(\text{死亡}|x), \quad x = (x_1, x_2, x_3) \quad x_1 = \text{年龄}, \quad x_2 = \text{性别} \in \{0,1\} \quad x_3 = \text{胆固醇}. \tag{3.30}$$

最简单的模型是考虑输入变量的线性组合：

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 = w^\top x, \quad x = (x_1, x_2, x_3) \tag{3.31}$$

但这不是概率！我们可以通过用 sigmoid 函数进行映射来修正这个问题：因此我们的模型为：

$$p(y|x) = \sigma(w^\top x), \quad \sigma(a) = \frac{1}{1+e^{-a}} : \text{逻辑函数} \tag{3.32}$$

直觉：这是试图拟合一个平面来区分两个事件 {死亡，不死亡}。

3.1 正式定义

设置：给定 $\mathcal{D} = ((x_1, y_1), ..., (x_n, y_n))$，$x_i \in \mathbb{R}^d$，$y_i \in \{0,1\}$。
模型：$y_i \sim \text{Ber}(\sigma(w^\top x_i))$，其中 y_i 是独立同分布的（i.i.d.）。

优点：

- 可解释性（= 模型参数具有可解释的意义，例如 $w_i > 0 \Rightarrow$ 死亡概率随年龄增加而增加）。$\Rightarrow$ 这就是它在医学等领域受欢迎的原因。
- 揭示哪些变量更具影响力。
- 参数数量少：仅有 $(d+1)$。这是一个简单的模型，在统计上易于训练。
- 计算上有效地估计 w。
- 容易扩展到多类分类。
- 形成更复杂模型（如神经网络和广义线性模型）的基础。

缺点：

- 作为一个简单模型，它的性能往往低于更复杂的方法。

3.2 最大似然估计

$$w_{\text{MLE}} = \arg\max_w p(\mathcal{D}|w), \quad p(\mathcal{D}|w) = \prod_{i=1}^{n} p(y_i|x_i, w), \tag{3.33}$$

设 $a_i = \sigma(w^\top x_i)$，那么 $p(\mathcal{D}|w) = \prod_{i=1}^n a_i^{y_i}(1-a_i)^{1-y_i}$，

$$\Rightarrow \mathcal{L}(w) = -\log p(\mathcal{D}|w) = -\sum_{i=1}^{n} [y_i \log a_i + (1-y_i)\log(1-a_i)] \to \text{二元交叉熵.} \tag{3.35}$$

3.3 迭代加权最小二乘法 Iterative reweighted least squares

回顾牛顿法：

$$w_{t+1} = w_t - H^{-1}g, \quad H = X^\top AX, \quad g = X^\top(a-y)$$

$$\Rightarrow w_{t+1} = w_t - (X^\top AX)^{-1}X^\top A[Xw_t - A^{-1}(a-y)]X^\top Ay \quad \text{假设} X^\top AX \text{ 可逆.} \tag{3.37}$$

我们可以重写为：

$$w_{t+1} = (X^\top AX)^{-1}X^\top A[Xw_t - A^{-1}(a-y)] = (X^\top AX)^{-1}X^\top Az_t, \tag{3.38}$$

其中 $z_t = Xw_t - A^{-1}(a-y)$。

回顾线性回归的解：$w_{\text{MLE}} = (X^\top X)^{-1}X^\top y = (X^\top X)^{-1}X^\top Ay$。那么 $w_{t+1} = (X^\top AX)^{-1}X^\top Az_t$ 是加权最小二乘问题的解，其中 A 是权重矩阵。

在实践中，并非使用逆矩阵 $H = X^\top AX$，通常解决系统 $Hu = f$ 使用共轭梯度法（CG）。如果遇到 H 不可逆的情况，可以切换回梯度下降法。

3.4 多类逻辑回归

模型 Model：

$$p(y = c|x, W) = \frac{\exp(w_c^\top x)}{\sum_{c'=1}^{C}\exp(w_{c'}^\top x)} \tag{3.39}$$

softmax 函数是逻辑函数在多类情况下的推广。这里 w_c 是 W 的第 c 列。

现在 y 是一个 One-Hot encoding 向量 $y = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^\top$，且 $y_{ic} := 1(y_i = c)$。

$$p(\mathcal{D}|w) = \prod_{i=1}^{n}\prod_{c=1}^{c} p(y_i = c|x_i, w),$$

$$\Rightarrow \log p(\mathcal{D}|w) = \sum_{i=1}^{n}\left(\sum_{c=1}^{c} y_{ic}w_c^\top x_i\right) - \log\left(\sum_{c'=1}^{c}\exp(w_{c'}^\top x_i)\right), \tag{3.41}$$

这是多类交叉熵损失。

梯度： 回顾 $a = \sigma(w^\top x)$。

$$\Rightarrow \log a = \log \sigma(w^\top x) = \log\left(\frac{1}{1+e^{-w^\top x}}\right) = -\log(1+e^{-w^\top x}), \tag{3.42}$$

以及

$$\log(1-a) = \log(1-\sigma(w^\top x)) = \log\left(1-\frac{1}{1+e^{-w^\top x}}\right) = \frac{-e^{-w^\top x}}{1+e^{-w^\top x}} = -w^\top x - \log\left(\frac{1}{1+e^{-w^\top x}}\right) \tag{3.43}$$

$$\frac{\partial}{\partial w}\log a = \frac{-xe^{-w^\top x}}{1+e^{-w^\top x}} = x(1-a) \tag{3.44}$$

$$\frac{\partial}{\partial w}\log(1-a) = -x + x(1-a) = -ax. \tag{3.45}$$

因此，

$$\nabla_w \mathcal{L}(w) = -\sum_{i=1}^{n} y_i x_{ij}(1-a_i) - (1-y_i)x_{ij}a_i \tag{3.46}$$

$$= -\sum_{i=1}^{n} y_i x_{ij}a_i - x_{ij}a_i + y_i x_{ij}a_i \tag{3.47}$$

$$= \sum_{i=1}^{n}(a_i - y_i)x_{ij} \tag{3.48}$$

在矩阵向量表示中：$\nabla_w \mathcal{L}(w) = x^\top(a-y)$。

注意：我们无法解析求解 w 因为在 $\nabla_w \mathcal{L}(w) = 0$ 中，w 以非线性方式出现。

Hessian：

$$\frac{\partial^2}{\partial w_k \partial w_j}\mathcal{L}(w) = \sum_{i=1}^{n} x_{ij}x_{ik}a_i(1-a_i) = \sum_{i=1}^{n} x_{ij}x_{ik}a_i(1-a_i) = z_j^\top A z_k.$$

我们可以写成这样是因为 $\sum_{i=1} x_{ij}x_{ik}a_i(1-a_i)$ 是二次型，其中 $z_j = (x_{1j}, ..., x_{nj})^\top$ 是设计矩阵 X 的第 j 列，以及

$$A = \begin{bmatrix} a_1(1-a_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & a_n(1-a_n) \end{bmatrix}$$

$$\Rightarrow \nabla_w^2 \mathcal{L}(w) = X^\top A X \Rightarrow \text{可以证明这是正半定的，因此} \mathcal{L}(w) \text{ 是凸的。}$$

将牛顿算法应用于这个问题，结果是一个称为迭代加权最小二乘法的迭代算法。

Chapter 4

深度神经网络：反向传播、过拟合与正则化、自动微分

1 神经网络

优点:

- 自适应特征 / 基函数（参数化）。由于数据和计算资源的丰富，最近取得了突破。
- 灵活的非线性回归模型，能够近似任何函数!

缺点:

- 似然函数不再是凸的。
- 在数据稀缺的情况下容易过拟合。

直觉：一组参数化的逻辑回归模型。

到目前为止，我们讨论的线性模型基于固定基函数的线性组合 $\varphi_j(x): y = f\left(\sum_{j=1}^{d} w_j \varphi_j(x)\right)$:

$$f: \begin{cases} \text{线性, } \varphi: \text{恒等} \to \text{线性回归,} \\ \text{线性, } \varphi: \text{非线性} \to \text{带基函数的线性回归,} \\ \text{逻辑, } \varphi: \text{恒等} \to \text{逻辑回归.} \end{cases} \tag{4.1}$$

我们现在的目标是使基函数 $\varphi_j(x)$ 依赖于参数，并允许在模型训练过程中调整这些参数和系数。这导致了基本的前馈神经网络（没有反馈连接，如 RNN）模型，可以描述为一系列函数变换 (functional transformation)。

i.e. $\mathrm{y} = \mathrm{w}^{\top}\varphi(x;\theta)$

第一层: $\mathrm{h}_q^{(1)} = f^{(1)}\left(\sum_{i=1}^{D} w_{qi}^{(1)} x_i + b_q^{(1)}\right), \quad q = 1, ..., Q^{(1)}$

以矩阵向量记法表示: $\mathrm{H}^{(1)} = f^{(1)}\left(XW^{(1)} + b^{(1)}\right)$

其中: $\begin{cases} H^{(1)} \in \mathbb{R}^{N\times Q^{(1)}}, \\ X \in \mathbb{R}^{N\times D}, \\ W^{(1)} \in \mathbb{R}^{D\times Q^{(1)}}, \\ b^{(1)} \in \mathbb{R}^{1\times Q^{(1)}} \end{cases}$

输出层为: $\mathrm{y} = \mathrm{f}^{(L)}\left(H^{(L-1)}W^{(L)} + b^{(L)}\right)$

其中: $\begin{cases} y \in \mathbb{R}^{N\times Q^{(L)}}, \\ H^{(L-1)} \in \mathbb{R}^{N\times Q^{(L-1)}}, \\ W^{(L)} \in \mathbb{R}^{Q^{(L-1)}\times Q^{(L)}}, \\ b^{(L)} \in \mathbb{R}^{1\times Q^{(L)}} \end{cases}$

给定数据: $\{(\mathrm{x}_1, y_1), ..., (x_N, y_N)\}$, $x_i \in \mathbb{R}^D$, $y_i \in \mathbb{R}^{Q^{(L)}}$，我们需要对网络架构做以下选择:

隐藏层的数量: L

每层的维度: $Q^{(1)}, ..., Q^{(L)}$

激活函数: sigmoid, tanh, ReLU 等。

2 最常见的激活函数

- 恒等: $f(x) = x$, $f'(x) = 1$, 范围 $(-\infty, +\infty)$, C^∞.
- 逻辑: $f(x) = \frac{1}{1+e^{-x}}$, $f'(x) = f(x)(1 - f(x))$, 范围 $(0, 1)$, C^∞.
- Tanh: $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$, $f'(x) = 1 - f(x)^2$, 范围 $(-1, +1)$, C^∞.
- ReLU: $f(x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0 \end{cases}$, 范围 $[0, +\infty)$, C^0.
- 指数: $f(a, x) = \begin{cases} a(e^x - 1), & x < 0, \\ x, & x \geq 0 \end{cases}$, $f'(x) = \begin{cases} f(a, x), & x < 0, \\ 1, & x \geq 0 \end{cases}$, 范围 $(-a, +\infty)$, $\begin{cases} C^1, & a = 1, \\ C^0, & \text{否则} \end{cases}$.
- 正弦: $f(x) = \sin(x)$, $f'(x) = \cos(x)$, 范围 $[-1, +1]$, C^∞.
- 高斯: $f(x) = e^{-x^2}$, $f'(x) = -2xe^{-x^2}$, 范围 $(0, +1]$, C^∞.

3 最常见的输出单元

- 线性: $y = H^{(L-1)}W^{(L)} + b^{(L)}$
- 逻辑: $y = \sigma\left(H^{(L-1)}W^{(L)} + b^{(L)}\right)$, $\sigma(a) = \frac{1}{1+e^{-a}} \rightarrow$ 二分类。
- Soft-max: $z = H^{(L-1)}W^{(L)} + b^{(L)}$, $\text{softmax}(z_j) = \frac{\exp(z_j)}{\sum_j \exp(z_j)}$, $i = 1, ..., N$, $j = 1, ..., C$ 对应多分类。

4 训练

$$p(y|x, \theta) = \mathcal{N}(y|f(x, \theta), \sigma^2) \tag{4.9}$$

$$\Rightarrow -\log p(y|x, \theta) = \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left[f(x_i; \theta) - y_i\right]^2 + \frac{N}{2} \log(2\pi\sigma^2). \tag{4.10}$$

特别地，我们感兴趣的是估计网络参数 θ。为此，只需最小化平方误差损失:

回归:

$$\mathbb{E}(\theta) = \frac{1}{2} \sum_{i=1}^{N} \left[f(x_i; \theta) - y_i\right]^2, \quad \theta^* = \arg\min \mathbb{E}(\theta) \tag{4.11}$$

二分类:

$$\mathbb{E}(\theta) = -\sum_{i=1}^{N} y_i \log f(x_i; \theta) + (1 - y_i) \log(1 - f(x_i; \theta)) \tag{4.12}$$

并执行梯度下降:

$$\theta_{n+1} = \theta_n - \eta \nabla \mathbb{E}(\theta) \tag{4.13}$$

5 过拟合与正则化

5.1 L_2 参数正则化（权重衰减 weight decay）

这是最常见的正则化形式。修改损失函数为:

$$\mathbb{E}(\theta) = \frac{1}{2}\sum_{i=1}^{n}[f(x_i;\theta) - y_i]^2 + \frac{\lambda}{2}w^\top w. \tag{4.14}$$

这里第一项对应于高斯分布，第二项表示对权重的先验: $p(w) = \mathcal{N}(0, \lambda^{-1}I)$。驱使权重更靠近原点。**注意:** 这可能与网络映射的缩放特性不一致（见 Bishop 5.5.1)。理想情况下，我们希望有一个在进行线性变换时是不变的正则化器（即对权重的缩放和偏差的位移不变)。这样一个正则化器为: $\frac{\lambda_1}{2}\sum_{w\in W_1} w^2 + \frac{\lambda_2}{2}\sum_{w\in W_2} w^2$。更一般地，我们可以考虑如下形式的先验:

$$p(w) \propto \exp\left(-\frac{1}{2}\sum_k \alpha_k ||w||_{2,k}^2\right), \quad \text{其中} \quad ||w||_{2,k}^2 := \sum_{j\in W_k} w_j^2. \tag{4.15}$$

然后，我们可以通过最大似然估计学习最优的 α_k（与 ARD 有关)。

5.2 L_1 参数正则化

修改损失函数为

$$\mathbb{E}(\theta) = \frac{1}{2}\sum_{i=1}^{n}[f(x_i;\theta) - y_i]^2 + \alpha||w||_1. \tag{4.16}$$

这里 $||w||_1 := \sum_j |w_j|$ 是 L_1 范数。它在模型参数中诱导稀疏性（对于足够大的 α)，即丢弃不需要的特征。

5.3 早停法

在训练大型模型时，我们经常会观察到训练误差和测试误差随时间稳步下降，但在某个时刻，测试误差开始增加。

5.4 丢弃法（Dropout）

标准的神经网络隐藏层为:

$$H^{(l)} = f(H^{(l-1)}W^{(l)} + b^{(l)}), \tag{4.17}$$

使用丢弃法后变为:

$$r_j^{(l)} \sim \text{Bernoulli}(p) \tag{4.18}$$

$$z^{(l)} = r^{(l)} \odot H^{(l-1)} \quad \text{（逐元素运算 elementwise）} \tag{4.19}$$

$$H^{(l)} = f(z^{(l)}W^{(l)} + b^{(l)}). \tag{4.20}$$

5.5 数据增强

在分类任务中，我们可以增加...

5.6 网络初始化: Xavier 初始化

假设我们有一个输入 X 和一个线性神经元，其随机权重为 W，输出为 y。那么：

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d \tag{4.1}$$

让我们来看看每一项的方差：

$$\mathrm{Var}[w_i x_i] = \mathbb{E}[x_i]^2 \mathrm{Var}(w_i) + \mathbb{E}[w_i]^2 \mathrm{Var}(x_i) + \mathrm{Var}(w_i)\mathrm{Var}(x_i). \tag{4.22}$$

如果我们的输入和权重均为零均值，则可以简化为：

$$\mathrm{Var}[w_i x_i] = \mathrm{Var}(w_i)\mathrm{Var}(x_i). \tag{4.23}$$

如果我们还假设 x_i 和 w_i 都是独立同分布的，那么我们可以计算 y 的方差：

$$\mathrm{Var}[y] = \mathrm{Var}[w_1 x_1 + w_2 x_2 + \cdots + w_d x_d] = n\,\mathrm{Var}[w_i]\mathrm{Var}[x_i]. \tag{4.24}$$

换句话说，输出的方差是输入的方差，乘以 $\mathrm{Var}[w_i]$。因此，如果我们希望输入的方差和输出的方差相同，则 $\mathrm{Var}[w_i]$ 应为 1。

$$\mathrm{Var}[w_i] = \frac{1}{d_{\mathrm{in}}} = \frac{1}{d_{\mathrm{out}}}. \tag{4.25}$$

如果我们现在对反向传播的信号进行相同的分析（见 Glorot + Bengio），我们同样会得到 $\mathrm{Var}[w_i] = \frac{1}{d_{\mathrm{out}}}$。由于这些约束仅在 $d_{\mathrm{in}} = d_{\mathrm{out}}$ 时才能满足（而这在大多数情况下并不成立），我们可以妥协地取平均值：

$$\mathrm{Var}[w_i] = \frac{2}{d_{\mathrm{in}} + d_{\mathrm{out}}}. \tag{4.26}$$

从而得到所谓的 Xavier 初始化：

$$W \sim \mathcal{U}\left(-\frac{\sqrt{6}}{\sqrt{d_{\mathrm{in}} + d_{\mathrm{out}}}}, \frac{\sqrt{6}}{\sqrt{d_{\mathrm{in}} + d_{\mathrm{out}}}}\right), \quad \text{或} \quad w \sim \mathcal{N}\left(0, \frac{2}{d_{\mathrm{in}} + d_{\mathrm{out}}}\right). \tag{4.27}$$

Chapter 5

图像分类与卷积神经网络

1 卷积神经网络 (CNNs/ConvNets)

与普通神经网络 (NNs) 非常相似：

- 由具有可微分权重和偏置的神经元构成。
- 每个神经元接收相同的输入，执行点积运算，通常后跟非线性变换。
- 前向传播和损失函数都是完全可微的。

⇒ 通过明确假设输入 "活 (live)" 在一个网格上（例如，规则采样的时间序列、图像等），CNNs 能够解决高维输入的神经网络的复杂性。这一假设使我们能够将某些结构和属性编码到架构中。

例如，考虑一幅分辨率为 $32 \times 32 \times 3$ 的单幅图像。则在全连接神经网络的第一层中，单独一个神经元将会有 $32 \times 32 \times 3 = 3072$ 个权重。

好吧，这听起来是可以管理的，但考虑一个更实际的 $200 \times 200 \times 3$ 的图像。这将对应于 $120,000$ 个权重，仅仅对于一个神经元！(⇒ 过拟合)。⇒ 卷积神经网络是指在至少一个层中使用卷积而不是一般矩阵乘法的神经网络。

2 卷积：一维的定义

假设我们给定一个在规则间隔上采样的时间序列 $x(t)$（x 可能是有噪声的）。为了滤除噪声，我们希望计算测量值的某种加权平均值。为此，我们可以选择一个权重函数 $w(s)$，并得到一个新的平滑函数 $s(t)$：

$$s(t) = \int_{-\infty}^{\infty} x(s)w(t-s)ds = \int_{-\infty}^{\infty} w(s)x(t-s)ds, \quad s(t) = (x * w)(t), \tag{5.1}$$

卷积是一个交换的线性运算 (commutative linear operation)。

2.1 一维的离散卷积

$$s(t) = (x * w)(t) = \sum_{s=-\infty}^{\infty} x(s)w(t-s) \tag{5.2}$$

2.2 二维图像中的离散卷积 (包含 $m \times n$ 像素)

卷积:

$$S(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) = \sum_m \sum_n K(m,n)I(i-m, j-n) \tag{5.3}$$

相关性:

$$S(i,j) = \sum_m \sum_n K(m,n)I(i+m, j+n) \tag{5.4}$$

3 使用 PyTorch 对图像进行分类的卷积神经网络 (CNN)

为什么交叉熵是分类错误的一个好的选择?

回顾交叉熵和 KL 散度的定义，假设 p 是真实分布，q 是你的模型分布:

$$H(p,q) = -\int p(x)\log(q(x))dx \tag{5.5}$$

$$KL[p||q] = \int p(x)\log\left(\frac{p(x)}{q(x)}\right)dx = \int p(x)\log(p(x))dx - \int p(x)\log(q(x))dx = \text{const} + H(p,q) \tag{5.6}$$

这意味着最小化交叉熵等价于最小化 KL 散度。换句话说，它告诉你你的替代模型正在寻找一种更好的方式来使预测密度与原始数据分布相匹配。

Chapter 6

循环神经网络与 LSTM

1 循环神经网络 recurrent neural network

到目前为止，我们所讨论的大多数预测任务都涉及简单的输出，例如实值或离散类别。然而，通常我们更关心的是预测更复杂的结构，例如图像或序列。如果输入和输出都是序列，我们称之为序列到序列的预测。

示例

- 语言建模：建模英语文本的分布。
- 语音转文本或文本转语音翻译。
- 标题生成：我们以图像为输入，想要生成该图像的自然语言描述。
- 机器翻译。

定义

我们有一个序列数据集 $\{y_t : t = 1, \ldots, T\}$，其中 $y_t \in \mathbb{R}^d$。我们的目标是将下一个序列值 $\hat{y}_t$ 建模为先前值 (previous value)（滞后 lags）$y_{t-1}, y_{t-2}, \ldots$ 的函数。假设两个滞后，这就转化为学习函数：$\hat{y}^t = f(y_{t-1}, y_{t-2})$。这定义了一个循环神经网络架构（即）：

$$\hat{y}_t = h_t V + c = \tanh(h_{t-1} W + y_{t-1} U + b) \tag{6.1}$$

$$h_{t-1} = \tanh(h_{t-2} W + y_{t-2} U + b) \tag{6.2}$$

$$h_{t-2} = 0. \tag{6.3}$$

注意到参数 W, U, b 是共享的。它们可以通过最小化均方误差损失进行训练：

$$\mathcal{L}(\theta) := \frac{1}{T-2} \sum_{t=3}^{\top} (y_t - \hat{y}_t)^2, \quad \theta := \{W, U, b, V, c\}. \tag{6.4}$$

示例

预测正弦波的动态，即：

$$y(t) = \sin(\pi t) \tag{6.5}$$

$y_t = f(y_{t-1}, y_{t-2})$，$y_t \in \mathbb{R}^{N\times D}$ 是一系列 d-值，y_{t-1}, y_{t-2} 是通过取滞后构建的序列（即）：

$$y_t := (y(2), y(3), \ldots, y(t)), \tag{6.6}$$
$$y_{t-1} := (y(1), y(2), \ldots, y(t-1)), \tag{6.7}$$
$$y_{t-2} := (y(0), y(1), \ldots, y(t-2)). \tag{6.8}$$

输入：$X \in \mathbb{R}^{L\times N\times D} : (y_{t-1}, y_{t-2})$。输出：$Y \in \mathbb{R}^{N\times D} : \hat{y}_t$。前向传播：

$$h_{t-2} = 0 \tag{6.10}$$
$$h_{t-1} = \tanh(h_{t-2}W + X[0,:,:]U + b) \tag{6.11}$$
$$h_t = \tanh(h_{t-1}W + X[1,:,:]U + b) \tag{6.12}$$
$$\hat{y}_t = h_t V + c \tag{6.13}$$

2 长短期记忆网络 (LSTM)

在 LSTM 中，隐藏单元 $h_t = \tanh(h_{t-1}W + y_{t-1}U + b)$ 被替换为：

$$h_t := o_t \odot \tanh(s_t) \quad \text{输出向量} \quad (y_{t-1}:\text{输入向量}) \to \text{输出门} \tag{6.14}$$
$$o_t = \sigma(h_{t-1}W_o + y_{t-1}U_o + b_o) \quad \text{输出门} \tag{6.15}$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t \quad \text{细胞状态} \tag{6.16}$$
$$\tilde{s}_t = \tanh(h_{t-1}W_s + y_{t-1}U_s + b_s) \tag{6.17}$$
$$i_t = \sigma(h_{t-1}W_i + y_{t-1}U_i + b_i) \quad \text{外部输入门} \tag{6.18}$$
$$f_t = \sigma(h_{t-1}W_f + y_{t-1}U_f + b_f) \quad \text{遗忘门} \tag{6.19}$$

参数 $\theta := \{W_o, U_o, b_o, W_s, U_s, b_s, W_i, U_i, b_i, W_f, U_f, b_f\}$ 在滞后间共享，并可以通过最小化 MSE 损失进行学习。

3 门控递归单元 (GRU)

一种参数更少的替代单元：

$$z_t = \sigma(h_{t-1}W_z + y_{t-1}U_z + b_z) \quad \text{更新门向量} \tag{6.20}$$
$$r_t = \sigma(h_{t-1}W_r + y_{t-1}U_r + b_r) \quad \text{重置门向量} \tag{6.21}$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tanh((r_t \odot h_{t-1})W_h + y_{t-1}U_h + b_h) \tag{6.22}$$

带有参数 $\theta := \{W_z, U_z, b_z, W_r, U_r, b_r, W_h, U_h, b_h\}$。

Chapter 7

监督学习的一般概念

1 一般框架

- 存在一个未知的分布 p 在 $\mathcal{X} \times \mathcal{Y}$ 上。
- 训练集 $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ 是来自 p 的 IID 样本。
- 假设是一个函数 $\hat{f} : \mathcal{X} \to \mathcal{Y}$。
- 学习算法是一个函数 $\mathcal{A} : S \to \hat{f}$。
- 假设空间 $\mathcal{F}$ 是 $\mathcal{A}$ 可访问的函数空间。

我们希望 f 在未来从 p 中抽取的样本上表现良好，其中“良好”的定义是基于损失函数 $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$。

我们怎么能确定能够找到一个好的 f，尤其是当 $\mathcal{F}$ 非常庞大时？这正是统计学习理论的研究内容。

2 三种误差的概念

- **训练误差或经验误差：** $\epsilon_{\text{emp}}[\hat{f}] = \frac{1}{m} \sum_{i=1}^{m} l(\hat{f}(x_i), y_i)$。这是我们可以实际测量的，也是许多学习算法尝试最小化的目标。→ 经验风险最小化（ERM）。
- **测试误差：** $\epsilon_{\text{test}}[\hat{f}] = \frac{1}{m'} \sum_{i=1}^{m'} l(\hat{f}(x'_i), y'_i)$
- **真实误差：** $\epsilon_{\text{true}}[\hat{f}] = \mathbb{E}_{(x,y)\sim p} l(f(x), y)$。这是理想算法所希望最小化的，但由于 p 是未知的，我们无法直接测量。

3 过拟合与欠拟合的权衡

一般来说，由于 $\hat{f}$ 是明确选择以最小化 $\epsilon_{\text{emp}}[\hat{f}]$，因此它往往会过于乐观地估计误差，即 $\epsilon_{\text{true}}[\hat{f}] > \epsilon_{\text{emp}}[\hat{f}]$ 和 $\epsilon_{\text{test}}[\hat{f}] > \epsilon_{\text{emp}}[\hat{f}]$。这称为过拟合。见图 7.3。问题是：我们如何避免过拟合？限制假设空间！但是多少合适？

假设空间应该足够大，但又不能过大。或者，我们需要添加正则化项。

4 正则化风险最小化（Regularized Risk Minimization, RRM）

在监督学习中找到过拟合和欠拟合之间适当平衡的最一般框架是 RRM：

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \left[\frac{1}{m} \sum_{i=1}^{m} l(f(x_i), y_i) + \lambda \Omega(f) \right]$$

其中

- $\mathcal{F}$: 假设空间。
- $l(\hat{y}, y)$: 损失函数。
- $\Omega : \mathcal{F} \to \mathbb{R}^+$: 正则化泛函。

通过调整正则化参数 λ，可以找到合适的折衷。两个好的例子是线性回归中的 Lasso 和 Ridge：

$$L_{\text{Lasso}} = \|y - X\beta\|_2{}^2 + \lambda \|\beta\|_1,$$

$$L_{\text{Ridge}} = \|y - X\beta\|_2{}^2 + \lambda \|\beta\|_2{}^2.$$

Lasso 在强制稀疏性方面表现良好，而 Ridge 也有机制来正则化参数的可行空间 (feasible space)。更多内容可以在 Tibshirani 的论文中找到。此外，参见图 7.1。

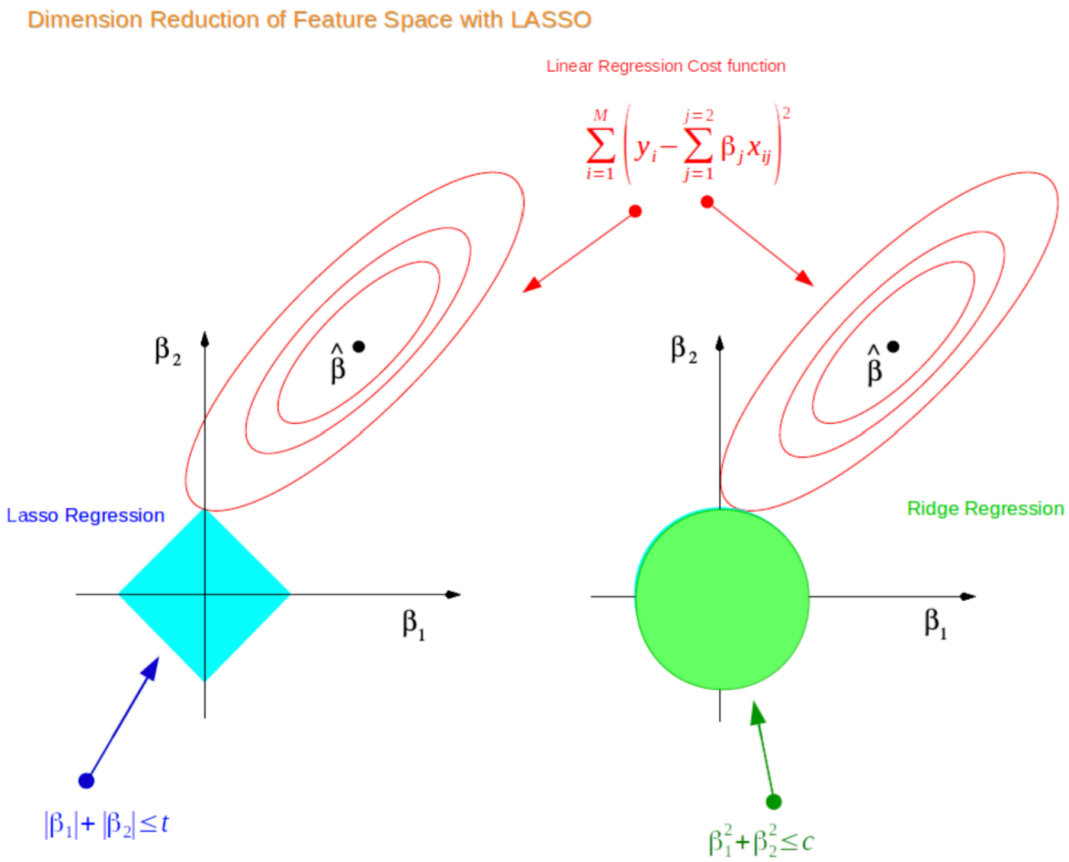

图 7.1: Enter Caption

5 统计学习理论

度量集中性 (Concentration of Measure)：我们能够学习到好的 $\hat{f}$ 的希望在于，对于任何固定的 $f \in \mathcal{F}$，无论选择训练集还是测试集，

$$\mathbb{E}(\epsilon_{\text{emp}}[\hat{f}]) = \mathbb{E}(\epsilon_{\text{test}}[\hat{f}]) = \mathbb{E}(\epsilon_{\text{true}}[\hat{f}]).$$

而且，随着训练和测试集大小的增加，这些量越来越集中于 $\mathbb{E}(\epsilon_{\text{true}}[\hat{f}])$。然而，一个非常不幸的训练集选择总是可能导致问题。

6 泛化界限 Generalization Bounds

统计学习理论证明了如下形式的界限：

$$\mathbb{P}(\epsilon_{\text{true}}[\hat{f}] > \epsilon_{\text{emp}}[\hat{f}] + \epsilon) < \delta$$

其中 ϵ 是一个复杂函数，依赖于 δ、函数类 $\mathcal{F}$ 的大小以及正则化的性质。这被称为概率近似正确（PAC）界限。例如，如果 $\mathcal{F}$ 的 VC 维数为 d，那么：

$$\mathbb{P}(\epsilon_{\text{true}}[\hat{f}] > \epsilon_{\text{emp}}[\hat{f}] + \sqrt{\frac{d}{m} d \left(\log\left(\frac{2m}{d} + 1 \right) \right) + \log\left(\frac{4}{\delta} \right)} < \delta.$$

实际上，即使是最好的此类界限往往也非常宽松。

7 交叉验证

保留集 Holdout set：统计学习理论中的泛化界限完全基于训练集，因此它们揭示了学习算法的基本性质。然而，它们通常是非常宽松的，对实际应用的指导性不高。

更实际的评估方法　估计 ϵ_{true} 的更实际的方法是将训练数据拆分为：

- 真实训练集，用于训练算法。
- 一个保留集 $\{(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)\}$，仅用于计算保留误差 $\epsilon_{\text{h.o.}}[\hat{f}] = \frac{1}{p} \sum_{i=1}^{p} l(\hat{f}(x_i), y_i)$。

保留误差是 $\epsilon_{\text{true}}[\hat{f}]$ 的无偏估计，即 $\mathbb{E}[\epsilon_{\text{h.o.}}[\hat{f}]] = \epsilon_{\text{true}}[\hat{f}]$。

7.1 k 折交叉验证

k 折交叉验证利用保留思想来设置学习算法的内部参数 θ（例如，k-NN 中的 k）：

- 将 θ 设置为某个值。
- 将训练集拆分为 k 个大致相等的部分 $S_1, S_2, \ldots, S_k$。
- 对于每个 i，在 $S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_k$ 上训练算法以获得 $\hat{f}_i$。
- 计算每个 $\hat{f}_i$ 在相应的 S_i 上的保留误差。
- 取平均以获得 $\epsilon_{\text{true}}[\hat{f}_\theta]$ 的估计。
- 在一系列 θ 值上迭代，最后将 θ 设置为最小化交叉验证误差的值。

Chapter 8

采样与不确定性量化

1 采样方法

- 给定 $p(x)$，进行样本抽取。
- 估计样本，学习 $p(x)$。
- 估计统计量。
- 进行贝叶斯推断。

为什么采样如此强大和有用?
近似: 它使我们能够近似期望值:

- 估计统计量。
- 后验推断。

采样: 从复杂分布中可视化典型抽样（然后可能对样本进行聚类?）
⇒ 为什么期望值?

- 任何概率都是期望值，例如，$p(x \in A) = \mathbb{E}[\mathbb{I}_{x\in A}], \mathbb{I}_{x\in A} := \begin{cases} 1, & x \in A, \\ 0, & x \notin A. \end{cases}$
- 近似对于不可解的 (intractable) 和或积分 (sums or integrals) 是必要的。许多和或积分可以写成期望值的形式。

优点: （mc、mcmc、IS 等）

- (1) 它们易于使用/实现/理解。
- (2) 它们具有很强的通用性。
- (3) 它具有良好的渐近理论保证（但在实践中可能效率低下）。

缺点:

- (1) 它们过于简单易用/实现/理解。经常不当使用。
- (2) 与精确或确定性近似相比，它们往往较慢（需要许多样本）。
- (3) 获得“良好/代表性”样本可能很困难/低效。
- (4) 评估该方法的性能可能很困难。

2 不同的场景

- 评估概率：给定 x，计算 $p(x) = \frac{\tilde{p}(x)}{Z_p}$，或未归一化的 $\tilde{p}(x)$。
- 从分布中抽样：给定 $p(x)$，生成一个代表性的样本 x。
- 评估统计量/矩：给定 $p(x)$，计算 $\mathbb{E}_{p(x)}[f(x)]$。
- 大挑战：x 是高维的，p、f 是复杂的。

3 蒙特卡洛近似

历史
这是一个简单的想法，但直到 1940 年代才建立了正式的数学基础。
恩里科 · 费米（因辐射研究获诺贝尔奖，1938 年，年仅 37 岁!），意大利物理学家。

- → 他在手动进行蒙特卡洛近似（非常繁琐)!（试图估计实验的结果。）
- 约翰 · 冯 · 诺依曼（美国最伟大的数学家之一)：(1940 年代）热衷于构建和设计计算机。
- ⇒ 这些人在二战期间都在洛斯阿拉莫斯工作，致力于开发核弹...
- 战后，在 1946 年，乌拉姆在生病的日子里想要估计/计算在纸牌游戏中获得完美“手牌”的概率。他无法进行分析，因此他考虑使用计算机和抽样进行近似。

这些人聚在一起，发展了蒙特卡洛近似的现代基础。
为什么叫做蒙特卡洛?
故事说乌拉姆有个叔叔喜欢纸牌游戏和赌场赌博。

示例

- 这个班级的学生的平均身高是多少? $\mathbb{E}[h] = \frac{1}{N}\sum_{i=1}^{N} h_i$。
- 中心城市的人的平均身高是多少?

  $$\mathbb{E}_{p\in C}[h(p)] \approx \frac{1}{|C|}\sum_{p\in C} h(p) \approx \frac{1}{S}\sum_{i=1}^{S} h(p_S),$$ 但我们不知道中心城市有多少人...

  对于在 C 中的随机调查的 S 个人，$p_s \sim$ 是从 C 中独立抽取的。
- 使用概率模型进行预测：$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$
  $\rightarrow p(x^*|D) = \int p(x^*|\theta, D)p(\theta|D)d\theta \approx \frac{1}{S}\sum_{s=1}^{S} p(x^*|\theta^{(s)}, D)$, $\theta^{(s)} \sim p(\theta|D)$。

4 重要性抽样 Importance Sampling（不是一种抽样方法!）

它是蒙特卡洛的扩展，用于近似难以处理的积分。
回想一下，这假设我们可以轻松地从 $p(x)$ 中抽样!
重要性抽样可以帮助我们处理无法从 $p(x)$ 生成样本的情况。但即使在可以从 $p(x)$ 中抽取样本的情况下，重要性抽样也可以帮助我们提高准确性（我们估计的收敛速率!）

设置 假设 $p(x)$ 是一个密度。然后：

$$\mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{p(x_i)}{q(x_i)},\ x_i \sim q(x) \tag{8.1}$$

$$\forall q(x) \text{ pdf such that } q(x) = 0 \Rightarrow p(x) = 0,\ \text{i.e., } p \text{ is absolutely continuous with respect to } q. \tag{8.2}$$

$$\Rightarrow \mathbb{E}_{p(x)}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)w(x_i) = \hat{\mu}_n^{IS},\ \text{where } x_i \sim q(x) \text{ serves as proposal distribution.} \tag{8.3}$$

且 $w(x_i) := \frac{p(x_i)}{q(x_i)} \rightarrow$ 重要性权重.

备注

1. $\mathbb{E}_{q(x)}[\hat{\mu}_n^{IS}] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{q(x)}\left[f(x_i)\frac{p(x_i)}{q(x_i)}\right] = \frac{1}{n}\sum_{i=1}^{n}\int f(x)\frac{p(x)}{q(x)}q(x)dx = \hat{\mu}_b$

   因此 IS 是无偏的、一致的，并且收敛速度为 $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$。

2. $\text{Var}[\hat{\mu}_n^{IS}] = \text{Var}\left[f(x)\frac{p(x)}{q(x)}\right]$，这与简单的 MC 方差估计不同，这暗示了如何使用 IS 来提高我们估计器的准确性，具体取决于我们如何选择 $q(x)$。

实际上，求解最优的理论值 $q^*(x)$ 是简单的，但在实践中可能很难从该分布中抽样。

场景

- (1) 不能从 p 抽样，使用 IS 来纠正从可处理的分布 q 抽样（然后选择 q 使其接近 p）。
- (2) 无论我们是否能够从 p 抽样，使用 IS 来改进基本的 MC 估计器。

5 蒙特卡洛近似

5.1 目标

使用样本来近似一个感兴趣的量（例如期望）。
$\mathbb{E}[f(x)]$，$x \in \mathbb{R}^d$，期望是难以处理的。

定义 如果 $x_1, ..., x_n \sim p$，那么：$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$ 是一个基本的蒙特卡洛估计器。同时 $\mathbb{E}_{x\sim p(x)}[f(x)] := \int f(x)p(x)dx$ 这只是样本均值。

备注

1. $\mathbb{E}[\hat{\mu}_n] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[f(x_i)] \rightarrow_{n\rightarrow\infty} \mathbb{E}[f(x)]$，因此 $\hat{\mu}_n$ 是一个无偏估计器。
2. $\hat{\mu}_n \rightarrow \mathbb{E}[f(x)]$ as $n \rightarrow \infty$(概率收敛：$\forall \epsilon > 0, p(|\hat{\mu}_n - \mathbb{E}[f(x)]| < \epsilon) \rightarrow 1$)。因此 $\hat{\mu}_n$ 是一个一致估计器。（即当 n 大时，我们得到正确答案的概率非常高，假设 $\text{Var}[f(x)] < \infty$，根据大数法则）。

5.2 样本均值的收敛性

- (3) $\text{Var}[\hat{\mu}_n] = \frac{1}{n}\text{Var}[f(x_i)] \rightarrow \frac{1}{n}\text{Var}[f(x)] \xrightarrow{n\rightarrow\infty} \frac{\sigma}{\sqrt{n}}\text{std}[f(x)]$，因此收敛速率为 $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$（无论 x 的维度如何！）。
- (4) 结合 (1) 和 (3) 我们可以得出：$\text{MSE}[\hat{\mu}_n] = \text{bias}^2 + \text{var} = \frac{1}{n}\text{Var}[f(x)] \xrightarrow{n\rightarrow\infty} 0$。

备注 尽管我们知道这个收敛速率，但知道实际误差是什么可能非常困难，因为我们不知道方差 $\mathrm{Var}[f(x)]$。(... 请记住，我们只是试图近似 $\mathbb{E}[f(x)]$)。
一个实际的限制：需要能够有效地从 $p(x)$ 中抽样！例如，使用 MC 近似 $\pi \approx 3.14$。

6 未归一化分布的重采样方法

我们可能只知道 $p(x)$ 或 $q(x)$ 的归一化常数。例如，后验推断：$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto p(D|\theta)p(\theta)$。

设置 假设 $p(x) = \frac{\tilde{p}(x)}{z_p}$，$q(x) = \frac{\tilde{q}(x)}{z_q}$，$\int \tilde{q}(x)dx = z_q > 0$ 和 $\int \tilde{p}(x)dx = z_p > 0$。一个替代的单位，参数更少：

$$\Rightarrow \mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx = \int f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}\frac{z_q}{z_p}q(x)dx. \tag{8.5}$$

$$\Rightarrow \mathbb{E}_{p(x)}[f(x)] = \int f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}\frac{z_q}{z_p}q(x)dx \approx \frac{z_q}{z_p}\frac{1}{n}\sum_{i=1}^{n} f(x_i)\tilde{w}(x_i),\ x_i \sim q(x),\ \tilde{w}(x) = \frac{\tilde{p}(x)}{\tilde{q}(x)}. \tag{8.6}$$

但我们仍然不知道归一化常数... 我们将使用蒙特卡洛近似来近似归一化常数的比率：

$$\frac{z_q}{z_p} = \frac{1}{z_p}\int \tilde{p}(x)dx = \frac{\int \frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)dx}{\mathbb{E}_{x\sim q(x)}\left[\frac{\tilde{p}(x)}{\tilde{q}(x)}\right]} \approx \frac{1}{n}\sum_{i=1}^{n}\tilde{w}(x_i). \tag{8.7}$$

$$\Rightarrow \mathbb{E}_{p(x)}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)\tilde{w}(x_i),\ x_i \sim q(x),\ \text{where } \tilde{w}(x_i) = \frac{\tilde{p}(x_i)}{\tilde{q}(x_i)}. \tag{8.8}$$

这里 $\tilde{w}(x_i)$ 扮演着" 近似重要性权重" 的角色。请记住，我们进行了两个近似以达到此目的！因此，预计其性能会比常规 IS 差。

如何选择好的重要性采样提议分布？例如：假设 $f(x)$ 是某项投资的回报，我们希望使用 IS 近似预期回报。
结论： 选择 $q(x)$ 在 $|f(x)p(x)|$ 较大时为大！
注意： 评估我们估计器的好坏可能很困难

7 拒绝采样

这是从一般分布生成精确样本的一种通用方法。(通常在低维度中有效...)

7.1 均匀情况

目标 Goal： 从某个复杂集合上生成均匀分布的样本。
我们假设可以评估指示函数 $\mathbb{1}_{(x\in A)} = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$ (可以达到一定的精度)。

基本思路： 从一个较大的简单集合 B 中抽取均匀样本。然后评估每个样本的 $\mathbb{1}_{(x\in A)}$，并选择是否接受或拒绝该样本。

这是一种在采样方法中常用的策略：从一个更简单的分布集合中抽样，检查一个确定性或随机条件，以决定是接受还是拒绝该样本。

命题： 如果 $A \subset B$，且 $y_1, y_2, \cdots \sim \mathrm{Uniform}(B)$ 并且独立同分布，$X = y_k$，其中 $k = \min\{k : y_k \in A\}$，那么 X 本身也是一个随机变量。

那么 $X \sim \mathrm{Uniform}(A)$。

7.2 非均匀情况（具有概率密度的情况）

目标 Goal：从复杂的概率密度 $p(x), x \in \mathbb{R}^d$ 中采样。
假设我们给定 $\tilde{p}(x)$，即未归一化的密度，$p(x) = \frac{\tilde{p}(x)}{z_p}, z_p > 0, z_p = \int \tilde{p}(x)dx$。

算法：

1. 选择一个概率密度函数 $q(x)$，使得：$\exists c > 0$，使得 $cq(X) \geq \tilde{p}(x), \forall x$（我们可以将 c 设为 $c = \max\left(\frac{\tilde{p}(x)}{q(x)}\right)$），并且 $q(x)$ 是易于抽样的。
2. 从 $q(x)$ 中抽样 x，然后从 Uniform$(0, cq(x))$ 中抽样 y（给定 x）。
3. 如果 $y \leq p(x)$，则 $z = x$，否则我们“拒绝” x，并返回步骤 #2。

输出 $z \sim p(x)$。
关键限制：在高维度中设计有效的提议分布非常困难...

8 马尔可夫链蒙特卡洛（MCMC）（20 世纪十大算法之一）

一种强大且通用的工具，用于近似期望和从复杂高维分布中抽样：

目标 Goal：从 $p(x)$ 中采样，或计算/近似 $\mathbb{E}_{x\sim p(x)}[f(x)], x \in \mathbb{R}^d$。
回顾基本的蒙特卡洛：$\mathbb{E}_{x\sim p(x)}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$, x_i 是从 $p(x)$ 中独立同分布抽样的。

$p(x)$ 可能太复杂而无法直接抽样。重要性采样或拒绝采样在高维中面临严重的困难，因为设计有效的提议分布变得极其困难。

8.1 MCMC 的直觉

想法：从某个 x_0 开始，找到高概率区域，然后通过随机移动在空间中导航/探索，同时保持在高概率区域附近。$p(x)$ 具有某种结构，质量集中在高维空间中的“流形 (manifold)”上。

最初的论文由 Metropolis 等人（1953 年）发表（Ulam 和 Metropolis 在 1949 年的论文中首次提出了使用马尔可夫链的想法）。

8.2 化学中的相图

在“箱子中的硬盘 (hard-disks in a box)”模型中，使用 MCMC 研究/建模材料的性质，尝试模拟该模型以计算期望 $\mathbb{E}_{x\sim p(x)}[f(x)], x \in \mathbb{R}^{2n}$，其中 $p(x)$ 是所有有效配置/状态的概率密度（例如，玻尔兹曼分布 $p(x) \propto e^{-E/kT}$，E 是状态能量，T 是热力学温度，k 是普朗克常数）。由于没有重叠约束和周期边界，这很复杂。

想法：从初始配置 x_0 开始，开始探索所有有效配置的空间。

如何探索？定义一些可接受的移动：在每一步/迭代中，选择一个粒子，绘制一个预定大小的框，并在该框中绘制一个均匀点。然后根据某种随机接受规则（Metropolis 规则）将粒子移动到该新位置。因此，在每次迭代中，我们可能会获得一个新配置 x_i。被接受配置的序列将用于近似所需的期望：$\frac{1}{n}\sum_{i=1}^{n} f(x_i)$。但现在 x_i 不是独立同分布的。

8.3 马尔可夫链的遍历定理 (Ergodic Theorem)

(G.D Birkhoff, 2010 年菲尔兹奖得主 Elan Lindsetrauss)

首先定义马尔可夫链!

一个动态系统是遍历的，如果它可以在有限时间内到达所有可能的状态，而与初始条件 x_0 无关。

回顾:

$$\text{Monte Carlo: } \mathbb{E}_{x\sim p(x)}[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \quad x_i \text{ 从}p(x) \text{ 中独立同分布抽样} \tag{8.9}$$

$$\text{MCMC: } \mathbb{E}_{x\sim p(x)}[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \quad x_i \text{ 从马尔可夫链中抽样} \tag{8.10}$$

我们希望证明，当 $n \to \infty$ 时，该估计是无偏且一致的。
这是通过遍历理论来完成的。

定理 1: 如果 $(x_0, x_1, \ldots, x_n)$ 是一个不可约（时间齐次）的离散马尔可夫链，具有平稳分布 π，则:

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) \xrightarrow{a.s.} \mathbb{E}_{x\sim\pi}[f(x)], \forall \text{有界函数} f : x \to \mathbb{R}.$$

如果进一步该链是非周期的，则 $p(X_n = x|X_0 = 0) \xrightarrow{n\to\infty} \pi(x)$ 。因此 x_n 是 $\pi(x)$ 的一个良好样本。

8.4 什么是马尔可夫链

$x_0 \to x_1 \to \cdots \to x_n$ 其中 $x_i \in \mathcal{X} \to \mathcal{X}$ 是一个可数集合，且 $x_i := (x_0, \ldots, x_n)$。

马尔可夫性质:

$$p(x_i|x_0, \ldots, x_{i-1}) = p(x_i|x_{i-1})$$

如果一个离散马尔可夫链是不可约的，具有平稳分布，并且是非周期的，则它是一个遍历马尔可夫链。

定义: 一个马尔可夫链 (x_i) 是（离散/时间）时间齐次的，如果 $p(x_{i+1} = b|x_i = a) = T_{ab}$，对于所有的 i 和 $a, b \in \mathcal{X}$，存在某个矩阵 T（对于连续时间的马尔可夫链可能是一个“无限”矩阵，我们会有一个转移核）。在这种情况下，转移概率不依赖于时间（索引 i）。
T 被称为马尔可夫链的转移矩阵，它是一个随机矩阵，即 $\sum_b T_{ab} = 1$（行和为一）。

定义: 一个概率质量函数 π 在 $\mathcal{X}$ 上是一个平稳/不变分布（相对于 T）如果 $\pi T = \pi$，即 $\sum_{a\in\mathcal{X}} \pi_a T_{ab} = \pi_b$，对所有的 $b \in \mathcal{X}$ 都成立。这让我们想起特征向量方程。实际上，π 被称为左特征向量，其特征值为 1。

定义: 一个马尔可夫链 (x_i) 是不可约的，如果 $\forall a, b \in \mathcal{X}$，$\exists t \geq 0$ 使得 $p(X_t = b|X_0 = a) > 0$。因此，无论从何处开始，我们都可以到达每个状态 b。

定义: 一个不可约的马尔可夫链 (x_i) 被称为非周期的，如果:

$$\forall a \in \mathcal{X}, \gcd\{t : p(X_t = a|X_0 = a) > 0\} = 1$$

其中 gcd 是最大公约数。$R_a = \{t : p(X_t = a|X_0 = b) > 0\}$ 是从状态 a 开始的返回到 a 的时间集合。

马尔可夫链示例

(1) $\mathcal{X}=\{1,2\}$ 转移矩阵 $T=\begin{bmatrix}p & q\\ p & q\end{bmatrix}$，它是一个随机矩阵，因为 $p+q=1$（行和为一）。该链是不可约的，因为 $p,q>0$，因此我们保证访问所有状态（如果 p 或 q 之一为零则不是这种情况）。结果发现它也是非周期的。它还有一个平稳分布 $\pi=(p,q)$，因为 $\pi T=\pi$，对于这样的 π。

(2) $\mathcal{X}=\{1,2,3,4\}$

转移矩阵 $T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0.5 & 0 & 0.5 & 0\\ 0 & 0.5 & 0 & 0.5\\ 0 & 0 & 0 & 1\end{bmatrix}$。这是带有反射边界的对称随机游走。

- 不可约性（是!）
- 非周期性（不是 $\rightarrow$ 事实上是周期性的。）

9 Metropolis 算法

设置 Setup：给定一个在状态集 $\mathcal{X}$ 上定义的概率质量函数 π（可数），以及一个函数 $f:\mathcal{X}\to\mathbb{R}$。

目标 Goal：近似从 π 中采样，或近似 $\mathbb{E}_{x\sim\pi}[f(x)]$（$\pi$ 和 f 可以非常复杂）。

方法：构造一个具有平稳分布 $\pi(x)$ 的马尔可夫链，使得马尔可夫链易于采样，然后依赖于遍历定理计算所需的近似值。

术语：

- 提议矩阵 (proposal materix) := 随机矩阵 (stochastic matrix)（即，所有条目都是非负的，所有行的和为 1）（$Q=Q_{ab},a,b\in\mathcal{X}$，其中 $Q_{ab}=Q(a,b)$）。
- $\pi(x)=\frac{\tilde{\pi}(x)}{z}$，$z>0$。Metropolis 算法仅需要我们评估未归一化的分布 $\tilde{\pi}(x)$。

算法：

1. 选择一个对称提议矩阵 Q（注意，对于 Metropolis-Hasting，Q 不需要是对称的）。
2. 初始化状态 $x_0\in\mathcal{X}$。
3. 对于 $i=0,1,2,\ldots,n-1$:
   - 从 $Q(x_i,x)$ 中采样 x（提议），即 $p(x|x_i)=Q(x_i,x)$。
   - 从均匀分布 $U(0,1)$ 中采样 u。
   - 根据随机 Metropolis 规则接受或拒绝：
     如果 $u<\frac{\tilde{\pi}(x)}{\tilde{\pi}(x_i)}$ 则 $x_{i+1}=x$，否则 $x_{i+1}=x_i$。
4. 输出 $x_0,\ldots,x_{n-1}$
   然后 $\mathbb{E}_{x\sim\pi}[f(x)]\approx\frac{1}{n}\sum_{k=0}^{n-1}f(x_k)$，并且 x_{n-1} 是从 $\pi(x)$ 的近似样本。

Metropolis 算法示例（“盒子中的硬盘模型”1953）一盒不重叠的刚性粒子（分子），固定半径。

- N 个粒子
- 周期性边界条件
- 相变的理论模型

目标是找到相应的状态方程：例如，对于理想气体：$PV = nRT$。

方法: 假设系统配置 $x = (r_1, s_1, \ldots, r_N, s_N) \in \mathbb{R}^{2N} \to [0,1]^{2N}$。
假设对所有可能状态的概率分布：

$$\pi(x) = \frac{1}{z} e^{-\frac{E(x)}{kT}} \cdot I_{\{x\}} \to \text{玻尔兹曼分布.} \tag{8.12}$$

$$\tilde{\pi}(x) = e^{-\frac{E(x)}{kT}} \cdot I_{\{x\}} \to \text{未归一化分布.} \tag{8.13}$$

其中 $I_{\{x\}}$ 意味着 x 必须是有效的。

- $e(x)$: 系统的能量（对于给定状态易于计算）。
- T: 系统的温度。
- k: 玻尔兹曼常数。
- z: 归一化常数（配分函数 partition function）$z = \int e^{-\frac{e(x)}{kT}} \pi(x)\, dx \to$ 非常高维的积分。

使用 Metropolis 算法来模拟该系统，以找到状态方程。我们希望计算关于 $\pi(x)$ 的期望。选择一个 T 和 N（体积已知），我们希望估计压力，这是一种关于 x 的函数，即 $\mathbb{E}_{x\sim\pi(x)}[f(x)]$。

Metropolis 算法

- 选择一个对称提议矩阵 Q: 随机选择一个第 k 个粒子，在盒子内均匀地采样一个点，即

  $$Q(x, x') = \frac{1}{N} \left\{(r', s') \forall i \neq k, |r_k - r'_k| \leq a, |s_k - s'_k| \leq a\right\} / C \tag{8.1}$$

  其中 C: 在 $2a \times 2a$ 盒子内的点的数量。对于 $x = x'$，$Q(x, x) = \frac{1}{C}$。
- 迭代：
  - 从提议分布中采样一个新状态。
  - 对于 $i = 0, ..., n-1$，采样一个 u，均匀分布 $U(0,1)$。
  - 评估：$\tilde{\pi}(x) = e^{-\frac{e(x)}{kT}} \mathbb{1}_{(x \text{ is valid})}$，并检查是否 $u < \frac{\tilde{\pi}(x)}{\tilde{\pi}(x_i)}$：$\begin{cases} \text{是的，} x_{i+1} = x \\ \text{否，} x_{i+1} = x_i \end{cases}$
  - $\to 123$

10 Gibbs 采样

假设我们有参数 $\theta = (\theta_1, \theta_2, \ldots, \theta_d)$ 和一些数据 x。

我们的目标是找到后验分布 $p(\theta|x)$。Gibbs 采样允许我们从后验中生成样本，如下所示：

1. 选择初始值 $\theta^{(i)} = (\theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_d^{(i)})$。

2. 采样：

$$\begin{aligned}
\theta_1^{(i+1)} &\sim p(\theta_1|\theta_2^{(i)}, \theta_3^{(i)}, \ldots, \theta_d^{(i)}, x) \\
\theta_2^{(i+1)} &\sim p(\theta_2|\theta_1^{(i)}, \theta_3^{(i)}, \ldots, \theta_d^{(i)}, x) \\
\theta_3^{(i+1)} &\sim p(\theta_3|\theta_1^{(i)}, \theta_2^{(i)}, \theta_4^{(i)}, \ldots, \theta_d^{(i)}, x) \\
&\vdots \\
\theta_d^{(i+1)} &\sim p(\theta_d|\theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_{d-1}^{(i)}, x)
\end{aligned}$$

3. 增加 $i = i + 1$，重复 M 次以绘制 M 个样本。

优点: 不需要调整任何参数（例如，相对于需要选择提议分布的 MCMC）。

缺点: 假设知道条件密度，这在实际中可能很难推导。

贝叶斯线性回归

$$y_i \sim \mathcal{N}(y_i|w_0 + w_1 x_i, \gamma^{-1}) \iff y_i = w_0 + w_1 x_i + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \gamma^{-1}).$$

$$\text{似然函数:} \quad \mathcal{L}(y_1, \ldots, y_n, x_1, \ldots, x_n|w_0, w_1, \gamma) = \prod_{i=1}^{n} \mathcal{N}(y_i|w_0 + w_1 x_i, \gamma^{-1}) \tag{8.15}$$

$$\text{先验分布:} \quad w = (w_0, w_1) \sim \mathcal{N}\left(\begin{bmatrix}\mu_0 \\ \mu_1\end{bmatrix}, \begin{bmatrix}\lambda_0^{-1} & 0 \\ 0 & \lambda_1^{-1}\end{bmatrix}\right), \quad \theta := \{w_0, w_1, \gamma\}, \quad \gamma \sim \text{Gamma}(\gamma|\alpha, \theta). \tag{8.16}$$

一般方法:

(i) 写出后验条件密度的对数形式。

(ii) 丢掉所有与当前采样变量无关的项。

(iii) 假装这是你所关注的变量的密度，而其他所有变量都是固定的。

对于 w_0 的 Gibbs 更新

$$p(w_0|w_1, \gamma, x, y) \propto p(y|x, w_0, w_1, \gamma)p(w_0) \tag{8.17}$$

其中 $p(y|x, w_0, w_1, \gamma)$ 是似然函数，$p(w_0) = \mathcal{N}(\mu_0, \gamma_0^{-1})$。

$$\log p(w_0 \mid w_1, \gamma, x, y) \propto -\frac{\gamma}{2}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2 - \frac{\lambda_0}{2}(w_0 - \mu_0)^2 \tag{8.18}$$

$$\propto \gamma \sum_{i=1}^{n}(y_i - w_1 x_i)w_0 - \frac{\lambda_0}{2}w_0^2 + \lambda_0 \mu_0 w_0 - \frac{\gamma}{2}n w_0^2 \tag{8.19}$$

展开并删除所有不依赖于 w_0 的项。这意味着：

$$p(w_0 \mid w_1, \gamma, x, y) \sim \mathcal{N}\left(w_0 \middle| \frac{\lambda_0 \mu_0 + \gamma \sum_{i=1}^{n}(y_i - w_1 x_i)}{\lambda_0 + n\gamma}, \quad (\lambda_0 + n\gamma)^{-1}\right) \tag{8.20}$$

对于 w_1 的 Gibbs 更新

对于 w_1 同样如此：

$$p(w_1 \mid w_0, \gamma, x, y) \propto p(y \mid x, w_0, w_1, \gamma) p(w_1) \tag{8.21}$$

其中 $p(y \mid x, w_0, w_1, \gamma)$ 是似然，$p(w_1) = \mathcal{N}(\mu_1, \gamma_1^{-1})$。我们仍然扩展对数似然并删除所有不依赖于 w_1 的项。

$$\log p(w_1 \mid w_0, \gamma, x, y) \propto -\frac{\gamma}{2} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2 - \frac{\lambda_1}{2}(w_1 - \mu_1)^2 \tag{8.22}$$

$$\propto \gamma \sum_{i=1}^{n} (y_i - w_0) x_i w_1 - \frac{\lambda_1}{2} w_1^2 + \lambda_1 \mu_1 w_1 - \frac{\gamma}{2} \sum_{i=1}^{n} w_0^2 x_i^2 \tag{8.23}$$

这意味着：

$$p(w_1 \mid w_0, \gamma, x, y) \sim \mathcal{N}\left(w_1 \middle| \frac{\lambda_1 \mu_1 + \gamma \sum_{i=1}^{n} (y_i - w_0) x_i}{\lambda_1 + \gamma \sum_{i=1}^{n} x_i^2}, \quad \left(\lambda_1 + \gamma \sum_{i=1}^{n} x_i^2 \right)^{-1} \right) \tag{8.24}$$

对 γ 的 Gibbs 更新

回顾一下：

$$\text{Gamma}(x|\alpha, \beta) \propto \beta^\alpha x^{\alpha-1} e^{-\beta x}, \quad \log \text{Gamma}(x|\alpha, \beta) \propto (\alpha - 1)\log(x) - \beta x. \tag{8.25}$$

$$p(\gamma|w_0, w_1, x, y) \propto p(y|x, w_0, w_1, \gamma) p(\gamma) \tag{8.26}$$

$$\propto \frac{n}{2}\log(\gamma) - \frac{\gamma}{2} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2 + (\alpha - 1)\log(\gamma) - \beta\gamma. \tag{8.27}$$

这意味着：

$$p(\gamma|w_0, w_1, x, y) \sim \text{Gamma}\left(\alpha - 1 + \frac{n}{2}, \quad \beta + \frac{1}{2} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2 \right). \tag{8.28}$$

Chapter 9

高斯过程、多输出高斯过程与多保真建模

1 高斯过程 (Recall)

先验 Prior

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2 I), \quad x \in \mathbb{R}^d, y \in \mathbb{R} \tag{9.1}$$

$$f(x) \sim \mathcal{GP}(0, K(x, x'; \theta)) \to \begin{bmatrix} f(x) \\ f(x') \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,x') \\ k(x',x) & k(x',x') \end{bmatrix} \right) \tag{9.2}$$

$$k(x, x'; \theta) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{i=1}^{d} \frac{(x_i - x_i')^2}{\theta_i^2} \right) \tag{9.3}$$

$$\theta := \{0, \sigma_n^2\} = \{\sigma_f^2, \theta_1, \ldots, \theta_d, \sigma_n^2\} \tag{9.4}$$

训练

数据 $\{X, y\}$，$X \in \mathbb{R}^{n \times d}, y \in \mathbb{R}^{n \times 1}$

$$p(y|x) = \mathcal{N}(0, k(X, X; \theta) + \sigma_n^2 I), \quad K = k(X, X; \theta) + \sigma_n^2 I \tag{9.5}$$

$$\Rightarrow -\log p(y|X) = \frac{1}{2} y^\top K^{-1} y + \frac{1}{2} \log |K| + \frac{n}{2} \log 2\pi \tag{9.6}$$

K 是 $n \times n$ 的，完全的，对称正定的 $\Rightarrow \mathcal{O}(n^3)$。梯度 $\nabla_\theta \log p(y|X)$ 可以通过解析方法或自动微分计算。

预测/后验（使用最优训练后的 θ^*）

$$\begin{bmatrix} f(x^*) \\ y \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(x^*,x^*) & k(x^*,X) \\ k(X,x^*) & k(X,X) \end{bmatrix} \right) \tag{9.7}$$

$$\Rightarrow \log p(f(x^*)|X, y) = \mathcal{N}(k(x^*)^\top K^{-1} y, \quad k(x^*, x^*) - k(x^*, X) K^{-1} k(X, x^*)) \tag{9.8}$$

其中 $\mu(x^*) = k(x^*, x) K^{-1} y$ 和 $\Sigma(x^*, x^*) = k(x^*, x^*) - k(x^*, X) K^{-1} k(X, x^*)$。

2 高斯过程（贝叶斯非参数方法用于非线性回归）

回顾线性回归与基函数 (basis function)：

$$y_i = w^\top \varphi(x_i), \quad y_i \in \mathbb{R},\, x_i \in \mathbb{R}^d,\, \varphi : \mathbb{R}^d \to \mathbb{R}^m, \quad i = 1, \ldots, n, \quad i.e.,\, \varphi(x) = (\varphi_1(x), \ldots, \varphi_m(x)) \tag{9.9}$$

通过最小化二次误差或最大似然估计，我们可以得到最优的 w^*：

$$w^* = (\phi^\top \phi)^{-1} \phi^\top y, \quad \text{其中} \phi = \begin{bmatrix} \varphi_1(x_1) & \ldots & \varphi_m(x_1) \\ \vdots & & \vdots \\ \varphi_1(x_n) & \ldots & \varphi_m(x_n) \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{9.10}$$

现在假设我们要在新的 x^* 处进行预测：

$$y^* = w^\top \varphi(x^*) = \left[(\phi^\top \phi)^{-1} \phi^\top y\right]^\top \varphi(x^*) = \varphi(x^*)^\top (\phi^\top \phi)^{-1} \phi^\top y \tag{9.11}$$

$K = \phi^\top \phi$ 的元素为：

$$K_{ij} = \sum_{m=1}^{m} \varphi_m(x_i)\varphi_m(x_j) \xrightarrow{m \to \infty} K(x, x') = \int \varphi_m(x)\varphi_m(x')dm \tag{9.12}$$

根据梅瑟尔定理 (Mercer's theorem)，$K = \phi^\top \phi$（参数化）和 $K(x, x') = \phi\phi^\top$（非参数化）是梅瑟尔核。前者，数据以一组参数的形式存储/概括；后者，数据作为参数的作用。

$$y = f(x) + \epsilon,\, \epsilon \sim \mathcal{N}(0, \sigma_n^2 I), \tag{9.13}$$

$$f(x) \sim \mathcal{GP}(f|0, K(x, x'; \theta)) \tag{9.14}$$

贝叶斯法则

$$p(f|y, X) = \frac{p(y|f, X)p(f)}{p(y|X)}, \quad p(y|X) = \int p(y|X, f)p(f) \tag{9.16}$$

$$X \in \mathbb{R}^{n \times d} = \begin{bmatrix} x_{11} & \ldots & x_{1d} \\ \vdots & & \vdots \\ x_{m1} & \ldots & x_{nd} \end{bmatrix}, \quad y \in \mathbb{R}^{n \times 1} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{9.17}$$

$$p(f|X) = \mathcal{N}(0, K) \Rightarrow \log p(f|X) = -\frac{1}{2} f^\top K^{-1} f - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi \tag{9.18}$$

$$p(y|f, X) = \mathcal{N}(f, \sigma_n^2 I) \xrightarrow[f]{\text{marginalize}} p(y|X) = \int p(y|X, f)p(f)df \tag{9.19}$$

$$\Rightarrow \log p(y|X) = -\frac{1}{2} \log |K + \sigma_n^2 I| - \frac{1}{2} y^\top (K + \sigma_n^2 I)^{-1} y - \frac{n}{2} \log 2\pi \tag{9.20}$$

3 多输出高斯过程回归

先验

$$x \in \mathbb{R}^d, y_1 \in \mathbb{R}, y_2 \in \mathbb{R} \tag{9.21}$$

$$y_1 = f_1(x) + \epsilon_1, \quad f_1(x) \sim \mathcal{GP}(0, K_1(x, x'; \theta_1)) \tag{9.22}$$

$$y_2 = f_2(x) + \epsilon_2, \quad f_2(x) \sim \mathcal{GP}(0, K_2(x, x'; \theta_2)) \tag{9.23}$$

$$c(x) \sim \mathcal{GP}(0, K_c(x, x'; \theta)) \tag{9.24}$$

$$\epsilon_1 \sim \mathcal{N}(0, \sigma_{n1}^2), \quad \epsilon_2 \sim \mathcal{N}(0, \sigma_{n2}^2) \tag{9.25}$$

训练

数据：$\{X_1, y_1\}, \{X_2, y_2\}$

$$p(y_1, y_2 | X_1, X_2) \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}K_{11}(X_1, X_1) & K_{12}(X_1, X_2)\\ K_{21}(X_2, X_1) & K_{22}(X_2, X_2)\end{bmatrix}\right) \tag{9.26}$$

$$K_{11}(X_1, X_1') := k_1(X_1, X_1'; \theta_1) + \sigma_{n1}^2 I \tag{9.27}$$

$$K_{12}(X_1, X_2) := k_c(X_1, X_2; \theta_c) \Rightarrow -\log p(y_1, y_2 | X_1, X_2) = \frac{1}{2} y^\top K^{-1} y + \frac{1}{2}\log|K| + \frac{n_1 + n_2}{2}\log 2\pi \tag{9.28}$$

$$K_{22}(X_2, X_2') := k_2(X_2, X_2'; \theta_2) + \sigma_{n2}^2 I, \quad \text{其中} y := \begin{bmatrix}y_1\\y_2\end{bmatrix} \tag{9.29}$$

预测

$$\begin{bmatrix}f_i(x^*)\\y_1\\y_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}K_{11}(x^*, x^*) & K_{11}(x^*, X_1) & K_{12}(x^*, X_2)\\ K_{11}(X_1, x^*) & K_{11}(X_1, X_1) & K_{12}(X_1, X_2)\\ K_{12}(X_2, x^*) & K_{12}(X_2, X_1) & K_{22}(X_2, X_2)\end{bmatrix}\right) \tag{9.30}$$

$$\Rightarrow p(f_i(x^*)|y) = \mathcal{N}((K_{11}(x^*, X_1)K_{12}(x^*, X_2))K^{-1}y, K_{11}(x^*, x^*) - k(x^*, X)K^{-1}k(x^*, X)^\top) \tag{9.31}$$

其中

$$k(x^*, X) = \begin{bmatrix}K_{11}(x^*, X_1)\\K_{12}(x^*, X_2)\end{bmatrix} \tag{9.32}$$

$$\begin{bmatrix}f_2(x^*)\\y_1\\y_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}K_{22}(x^*, x^*) & K_{12}(x^*, X_1) & K_{22}(x^*, X_2)\\ K_{12}(X_1, x^*) & K_{11}(X_1, X_1) & K_{12}(X_1, X_2)\\ K_{22}(X_2, x^*) & K_{12}(X_2, X_1) & K_{22}(X_2, X_2)\end{bmatrix}\right) \tag{9.33}$$

$$\Rightarrow p(f_2(x^*)|y) = \mathcal{N}((K_{12}(x^*, X_1)K_{22}(x^*, X_2))K^{-1}y, K_{22}(x^*, x^*) - k(x^*, X)K^{-1}k(x^*, X)^\top) \tag{9.34}$$

其中

$$k(x^*, X) = \begin{bmatrix}K_{12}(x^*, X_1)\\K_{22}(x^*, X_2)\end{bmatrix} \tag{9.35}$$

4 多保真 (Multi-fidelity) 高斯过程回归

先验

数据 $x \in \mathbb{R}^d, y_L \in \mathbb{R}, y_H \in \mathbb{R}$。

$$y_L = f_L(x) + \epsilon_L, \quad f_L(x) \sim \mathcal{GP}(0, K_L(x, x'; \theta_L)) \tag{9.37}$$

$$y_H = f_H(x) + \epsilon_H, \quad f_H(x) = \rho f_L(x) + \delta(x) \tag{9.38}$$

$$\delta(x) \sim \mathcal{GP}(0, K_\delta(x, x'; \theta_H)), \quad \delta(x) \perp f_L(x) \tag{9.39}$$

其中

$$\theta := \{\sigma_f^2, \theta_1^L, \ldots, \theta_d^L, \sigma_{fH}^2, \theta_1^H, \ldots, \theta_d^H, \rho, \sigma_{nL}^2, \sigma_{nH}^2\}. \tag{9.40}$$

训练

数据：$\{x_L, y_L\}, \{x_H, y_H\}$，$n_L \gg n_H$。

$$p(y_L, y_H | X_L, X_H) \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix} K_{LL}(X_L, X_L'; \theta_{LL}) & K_{LH}(X_L, X_H'; \theta_{LH}) \\ K_{HL}(X_H, X_L'; \theta_{LH}) & K_{HH}(X_H, X_H'; \theta_{HH}) \end{bmatrix}\right) \tag{9.40}$$

$$K_{LL}(X_L, X_L'; \theta_{LL}) = k_L(X_L, X_L'; \theta_L) + \sigma_{nL}^2 I, \quad \theta_{LL} = \{\theta_L, \sigma_{nL}^2\} \tag{9.41}$$

$$K_{LH}(X_L, X_H'; \theta_{LH}) = \rho k_L(X_L, X_H'; \theta_L), \quad \theta_{LH} = \{\theta_L, \rho\} \tag{9.42}$$

$$K_{HH}(X_H, X_H'; \theta_{HH}) = \rho^2 K_L(X_H, X_H'; \theta_L) + k_H(X_H, X_H'; \theta_H) + \sigma_{nH}^2 I, \quad \theta_{HH} = \{\theta_L, \theta_H, \sigma_{nH}^2, \rho\} \tag{9.43}$$

$$\Rightarrow -\log p(y_L, y_H | X_L, X_H) = \frac{1}{2} y^\top K^{-1} y + \frac{1}{2} \log |K| + \frac{n_L + n_H}{2} \log 2\pi \tag{9.44}$$

其中 $y := \begin{bmatrix} y_L \\ y_H \end{bmatrix}$。

预测

$$\begin{bmatrix} f_L(x^*) \\ \begin{bmatrix} y_L \\ y_H \end{bmatrix} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix}, \begin{bmatrix} K_{LL}(x^*, x^*) & K_{LL}(x^*, X_L) & K_{LH}(x^*, X_H) \\ K_{LL}(X_L, x^*) & K_{LL}(X_L, X_L) & K_{LH}(X_L, X_H) \\ K_{LH}(X_H, x^*) & K_{LH}(X_H, X_L) & K_{HH}(X_H, X_H) \end{bmatrix}\right) \tag{9.45}$$

$$\Rightarrow p(f_L(x^*)|y) = \mathcal{N}((K_{LL}(x^*, X_L) K_{LH}(x^*, X_H)) K^{-1} y, K_{LL}(x^*, x^*) - k(x^*, X) K^{-1} k(x^*, X)^\top) \tag{9.46}$$

$$\begin{bmatrix} f_H(x^*) \\ \begin{bmatrix} y_L \\ y_H \end{bmatrix} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix}, \begin{bmatrix} K_{HH}(x^*, x^*) & K_{LH}(x^*, X_L) & K_{HH}(x^*, X_H) \\ K_{LH}(X_L, x^*) & K_{LL}(X_L, X_L) & K_{LH}(X_L, X_H) \\ K_{HH}(X_H, x^*) & K_{LH}(X_H, X_L) & K_{HH}(X_H, X_H) \end{bmatrix}\right) \tag{9.47}$$

$$\Rightarrow p(f_H(x^*)|y) = \mathcal{N}((K_{LH}(x^*, X_L) K_{HH}(x^*, X_H)) K^{-1} y, K_{HH}(x^*, x^*) - k(x^*, X) K^{-1} k(x^*, X)^\top) \tag{9.48}$$

5 高斯过程分类（双分类情况 Binary case）

$$\pi(x) := p(y = +1|x) = \sigma(f(x)), \quad f(x) \sim \mathcal{GP}(0, K(x, x'; \theta)) \tag{9.49}$$

给定 $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}, x_i \in \mathbb{R}^d, y_i \in \{+1, -1\}, i = 1, \ldots, n$ 我们希望预测 $\pi(x^*)$。

贝叶斯规则

$$p(f|X,y) = \frac{p(y|f)p(f|X)}{p(y|X)} \tag{9.50}$$

由于分母与 f 无关，最大后验估计对应于最大化 $\log p(y|f) + \log p(f|X)$。由于后验不可求解，我们将通过拉普拉斯近似寻求高斯近似。为此：

$$\psi(f) := \log p(y|f) + \log p(f|X) = \log p(y|f) - \frac{1}{2} f^\top K^{-1} f - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi \tag{9.51}$$

$$\nabla_f \psi(f) = \nabla_f \log p(y|f) - K^{-1} f \tag{9.52}$$

$$\nabla_f^2 \psi(f) = \nabla_f^2 \log p(y|f) - K^{-1} = -W - K^{-1} \tag{9.53}$$

其中 $W := -\nabla_f^2 \log p(y|f)$ 是对角 Hessian 矩阵，因为似然函数是分解的 (factorized)（即数据 y_i 在给定 f_i 的条件下是独立的）。

训练

对于训练，我们需要计算边际似然的拉普拉斯近似：

$$p(y|X) = \int p(y|f)p(f|X)df = \int e^{\psi(f)} df \tag{9.63}$$

这里的 $\psi(f)$ 代表对数边际似然的对数。

在高斯过程分类中，推导出的后验分布通过拉普拉斯近似可以得到有效的样本和推断。

具体步骤

1. **选择初始值**: 设定初始的 f 值，可以随机选择或基于某个先验知识。
2. **计算梯度**: 通过 $\nabla_f \psi(f)$ 计算当前点的梯度。
3. **更新**: 根据牛顿法更新 f 的值：

$$f_{m+1} = (K^{-1} + W)^{-1}(W f_m + \nabla_f \log p(y|f_m))$$

4. **收敛检查**: 检查更新后的 f 是否收敛，通常使用 $\|f_{m+1} - f_m\| < \epsilon$ 来判断。
5. **计算后验分布**: 使用最后的 f 值计算后验分布。

模型选择与超参数优化

为优化模型性能，我们通常会使用交叉验证来选择超参数，如核函数的参数和噪声的标准差。通过最小化边际似然来确定超参数的最优值：

$$\theta^* = \arg\max_\theta p(y|X;\theta) \tag{9.64}$$

这可以通过梯度下降方法来实现，或者使用更复杂的贝叶斯优化技术来自动选择超参数。

应用实例

在实际应用中，高斯过程分类被广泛应用于模式识别、医学诊断等领域，尤其是在处理小样本、高维数据的场景中具有显著优势。

总结

高斯过程为我们提供了一种灵活且强大的工具，通过结合先验知识与数据进行贝叶斯推断，能够有效地进行非线性回归和分类任务。同时，借助拉普拉斯近似和适当的超参数选择策略，我们能够提升模型的泛化能力和预测性能。

在模式 $\hat{f}$ 周围局部使用 $\psi(f)$ 的泰勒展开式，我们得到：

$$\psi(f) \approx \psi(\hat{f}) - \frac{1}{2}(f - \hat{f})^\top A(f - \hat{f}) \tag{9.64}$$

因此，可以得到边际似然 $p(y|X)$ 的近似值：

$$p(y|X) \approx q(y|X) = \exp(\psi(\hat{f})) \int \exp\left[-\frac{1}{2}(f - \hat{f})A(f - \hat{f})\right] df \tag{9.65}$$

高斯积分可以通过分析获得：

$$-\log q(y|X) = \frac{1}{2} f^\top K f - \log p(y|\hat{f}) + \frac{1}{2}\log|B|. \tag{9.66}$$

Chapter 10

贝叶斯优化与主动学习

1 主动学习 Active Learning

给定一个训练好的高斯过程（GP）模型，其预测后验分布为：

$$p(f(x^*)|X, y, x^*) = \mathcal{N}(\mu(x^*), \Sigma(x^*)) \tag{10.1}$$

我们可以获取新的数据点，以最小化后验不确定性：

$$x_{n+1} = \arg\max_x \Sigma(x) \tag{10.2}$$

其中

$$\Sigma(x) := K(x, x) - K(x, X)(K + \sigma_n^2 I)^{-1} K(x, X) \tag{10.3}$$

2 贝叶斯优化

$$x_{n+1} = \arg\max_x \alpha(x; \mathcal{D}) \tag{10.4}$$

2.1 终止准则

- 最大迭代次数
- 连续样本之间的距离，即 $\|x_{n+1} - x_n\| < \epsilon$
- 获取函数的阈值，即 $\max \alpha(x; \mathcal{D}) < \epsilon$

2.2 从高斯过程采样

$$S(x) = \mu(x) + Lz, \quad \text{其中}\Sigma(x) = LL^T, \quad \text{且}z \sim \mathcal{N}(0, I) \tag{10.5}$$

2.3 贝叶斯优化获取函数

2.3.1 改进的概率 Probability of improvement

设 $f_{\min} := \min(x)$，即迄今为止观察到的最小值。我们希望在最有可能改善该值的点上评估 $f(x)$。这对应于以下效用函数 (utility function)：

$$u(x) = \begin{cases} 0, & f(x) > f_{\min}, \\ 1, & f(x) \leq f_{\min}. \end{cases} \tag{10.6}$$

改进的概率获取函数 (probability of improvement acquisition function) 是期望效用：

$$\alpha_{PI}(x; \mathcal{D}) = \mathbb{E}[u(x)|x, \mathcal{D}] = \int_{-\infty}^{f_{\min}} \mathcal{N}(f|\mu(x), K(x,x))\, df = \Phi(f_{\min}|\mu(x), K(x,x)) \tag{10.7}$$

其中 $\Phi(f_{\min}|\mu(x), K(x,x))$ 是高斯累积分布函数（cdf)。

2.3.2 期望改进

$$u(x) = \max\{0, f_{\min} - f(x)\} \tag{10.8}$$

然后，

$$\alpha_{EI} = \mathbb{E}[u(x)|x, \mathcal{D}] = \int_{-\infty}^{f_{\min}} (f_{\min} - f)\mathcal{N}(f|\mu(x), K(x,x))\, df \tag{10.9}$$

$$= (f_{\min} - \mu(x))\phi(f_{\min}|\mu(x), K(x,x)) + K(x,x)\mathcal{N}(f_{\min}|\mu(x), K(x,x)) \tag{10.10}$$

其中 $\phi(f_{\min}|\mu(x), K(x,x))$ 是正态分布的概率密度函数（pdf)，而 $\mathcal{N}(f_{\min}|\mu(x), K(x,x))$ 是正态分布的概率密度函数（pdf)。

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha(x, \mathcal{D}) \tag{10.11}$$

2.4 熵搜索

我们寻求最小化对最优值位置的不确定性：

$$x^* = \arg\min_{x \in \mathcal{X}} f(x) \tag{10.12}$$

我们对 f 的信念会引导出对 x^* 的分布，即 $p(x^*|\mathcal{D})$，然而这个分布没有封闭形式 (close form)。

熵搜索寻求评估点，以便最小化由 $p(x^*|\mathcal{D})$ 导致的熵。效用度量对应于给定新测量后熵的减少：

$$u(x) = \mathbb{H}[x^*|\mathcal{D}] - \mathbb{H}[x^*|\mathcal{D}, x, f(x)] \tag{10.13}$$

$$\alpha_{ES}(x; \mathcal{D}) = \mathbb{E}[u(x)|x, \mathcal{D}] \tag{10.14}$$

为了计算这个，需要进行一系列的近似。

Chapter 11

概率科学计算：高斯过程回归与微分方程的结合

1 高斯过程与微分方程的结合

1.1 Setup

给定一个线性算子：$\mathcal{L}_x u(x) = f(x)$ 和数据 $\{x_u, y_u\}, \{x_f, y_f\}$, $x \in \mathbb{R}^d$, $y_u, y_f \in \mathbb{R}$。

- $\{x_u, y_u\}$ 是 $u(x)$ 的少量测量（可以是边界或初始数据，也可以不是），$y_u = u(x) + \epsilon_u$。
- $\{x_f, y_f\}$ 是 $f(x)$ 的少量测量（$u(x)$ 和 $f(x)$ 都是未知的黑箱函数），$y_f = f(x) + \epsilon_f$。

1.2 模型

$$u(x) \sim \mathcal{GP}(0, K_{uu}(x, x'; \theta)) \tag{11.1}$$

$$f(x) \sim \mathcal{GP}(0, K_{ff}(x, x'; \theta)), \quad \text{with } K_{ff}(x, x'; \theta) = \mathcal{L}_x \mathcal{L}_{x'} K_{uu}(x, x'; \theta) \tag{11.2}$$

$$\epsilon_u \sim \mathcal{N}(0, \sigma_{n_u}^2), \quad \epsilon_f \sim \mathcal{N}(0, \sigma_{n_f}^2), \quad \theta := \{\sigma_f^2, \theta_1, ..., \theta_d, \sigma_{n_u}^2, \sigma_{n_f}^2\} \tag{11.3}$$

1.3 训练

$$\begin{bmatrix} y_u \\ y_f \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{uu} & K_{uf} \\ K_{uf} & K_{ff} \end{bmatrix}\right), \quad X = \begin{bmatrix} x_u \\ x_f \end{bmatrix} \tag{11.4}$$

并且

$$K_{uu} = K_{uu}(x_u, x_u'; \theta) + \sigma_{n_u}^2 I \tag{11.5}$$

$$K_{uf} = \mathcal{L}_{x'} K_{uu}(x_u, x_f'; \theta) \Rightarrow -\log p(y|X) = \frac{1}{2} y^\top K^{-1} y + \frac{1}{2} \log |K| + \frac{n_u + n_f}{2} \log 2\pi \tag{11.6}$$

$$K_{ff} = \mathcal{L}_x \mathcal{L}_{x'} K_{uu}(x_f, x_f'; \theta) + \sigma_{n_f}^2 I \tag{11.7}$$

1.4 预测

$$\begin{bmatrix} u(x^*) \\ y_f \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{uu}(x^*, x^*) & K_{uu}(x^*, X) \\ K_{uu}(x^*, X)^\top & K \end{bmatrix}\right) \tag{11.8}$$

$$\Rightarrow p(u(x^*)|X,y,x^*)=\mathcal{N}(K_{uu}(x^*,X)K^{-1}y,K_{uu}(x^*,x^*)-k_{uu}(x^*,X)K^{-1}k_{uu}(X,x^*)) \tag{11.9}$$

其中 $\mu_f(x^*)=K_{uu}(x^*,X)K^{-1}y$ 和 $\Sigma_u(x^*)=K_{uu}(x^*,x^*)-k_{uu}(x^*,X)K^{-1}k_{uu}(X,x^*)$。

同样地，

$$\begin{bmatrix}f(x^*)\\y_f\end{bmatrix}\sim\mathcal{N}\left(0,\begin{bmatrix}K_{ff}(x^*,x^*) & K_{fu}(x^*,X)\\K_{fu}(x^*,X)^\top & K\end{bmatrix}\right) \tag{11.10}$$

$$\Rightarrow p(f(x^*)|X,y,x^*)=\mathcal{N}(K_{fu}(x^*,X)K^{-1}y,K_{ff}(x^*,x^*)-k_{fu}(x^*,X)K^{-1}k_{fu}(X,x^*)) \tag{11.11}$$

2 线性微分方程的机器学习

2.1 Setup

给定数据 $\{x_u,y_u\},\{x_f,y_f\}$，我们希望学习一个模型：

$$y_u=u(x)+\epsilon_u\quad u(x)\sim\mathcal{GP}(0,K(x,x;\theta)) \tag{11.12}$$

$$y_f=f(x)+\epsilon_f\quad f(x)\sim\mathcal{GP}(0,g(x,x;\lambda)) \tag{11.13}$$

$$\mathcal{L}_x^a u(x)=f(x)\Rightarrow g(x,x';\theta,\lambda)=\mathcal{L}_x^a\mathcal{L}_{x'}^a K(x,x';\theta) \tag{11.14}$$

参数：$\theta=\{\sigma_f^2,\theta_1,...,\theta_d,\sigma_{n_u}^2,\sigma_{n_f}^2\}$

2.2 训练

$$\theta^*=\arg\min-\log p(y|X)=\frac{1}{2}\log|K|+\frac{1}{2}y^\top K^{-1}y+\frac{n_u+n_f}{2}\log 2\pi \tag{11.15}$$

$$y=\begin{bmatrix}y_u\\y_f\end{bmatrix},\quad X=\begin{bmatrix}x_u\\x_f\end{bmatrix},\quad y\sim\mathcal{N}\left(0,\begin{bmatrix}K_{uu} & K_{uf}\\K_{uf} & K_{ff}\end{bmatrix}\right) \tag{11.16}$$

并且

$$K_{uu}=K_{uu}(x_u,x_u';\theta)+\sigma_{n_u}^2 I \tag{11.17}$$

$$K_{uf}=\mathcal{L}_x K_{uu}(x_u,x_f';\theta) \tag{11.18}$$

$$K_{ff}=\mathcal{L}_x\mathcal{L}_{x'}K_{uu}(x_f,x_f';\theta) \tag{11.19}$$

2.3 预测

$$\begin{bmatrix}u(x^*)\\y_u\\y_f\end{bmatrix}\sim\mathcal{N}\left(0,\begin{bmatrix}K(x^*,x^*) & [K(x^*,x_u) & \mathcal{L}_x K(x^*,x_f)]\\\begin{bmatrix}K(x_u,x^*)\\\mathcal{L}_x K(x_f,x^*)\end{bmatrix} & & K\end{bmatrix}\right) \tag{11.20}$$

$$\Rightarrow p(u(x^*)|X,y,x^*)=\mathcal{N}(K(x^*,x_u)\mathcal{L}_{x'}K(x^*,x_f)K^{-1}y,K(x^*,x^*)-k(x^*,X)K^{-1}k(X,x^*)) \tag{11.21}$$

并且，

$$\begin{bmatrix}f(x^*)\\y_u\\y_f\end{bmatrix}\sim\mathcal{N}\left(0,\begin{bmatrix}\mathcal{L}_x\mathcal{L}_{x'}K(x^*,x^*) & [\mathcal{L}_x K(x^*,x_u) & \mathcal{L}_x K(x^*,x_f)]\\\begin{bmatrix}\mathcal{L}_x K(x_u,x^*)\\\mathcal{L}_x K(x_f,x^*)\end{bmatrix} & & K\end{bmatrix}\right) \tag{11.22}$$

$$\Rightarrow p(f(x^*)|X,y,x^*)=\mathcal{N}(\mathcal{L}_xK(x^*,x_u)\mathcal{L}_x\mathcal{L}_{x'}K(x^*,x_f)K^{-1}y,\quad \mathcal{L}_x\mathcal{L}_{x'}k(x^*,x^*)-k(x^*,X)K^{-1}k(X,x^*)) \tag{11.23}$$

其中我们得出结论 $\mu_u(x^*)=[K(x^*,x_u)\mathcal{L}_{x'}K(x^*,x_f)]K^{-1}y,\quad \Sigma_u(x^*)=k(x^*,x^*)-(x^*,X)K^{-1}(X,x^*)$,
$\mu_f(x^*)=[\mathcal{L}_xK(x^*,x_u)\mathcal{L}_x\mathcal{L}_{x'}K(x^*,x_f)]K^{-1}y$ 和 $\Sigma_f(x^*)=\mathcal{L}_x\mathcal{L}_{x'}K(x^*,x^*)-k(x^*,X)K^{-1}k(X,x^*)$。

2.4 一个详细示例：一维欧拉-伯努利梁

$$EI\frac{d^4}{dx^4}u(x)=f(x),\quad \text{边界条件}\quad \begin{cases}u(0)=0\\ u_x(0)=0\\ u_{xx}(1)=0\\ u_{xxx}(1)=0\end{cases} \tag{11.24}$$

在这种情况下，我们可以定义微分算子:

$$\mathcal{L}_xu(x)=f(x),\quad \mathcal{L}_x:=EI\frac{\partial^4}{\partial x^4}=\frac{d^4}{dx^4} \tag{11.25}$$

模型 $u(x)\sim\mathcal{GP}(0,K(x,x';\theta))\quad\Rightarrow f(x)\sim\mathcal{GP}(0,g(x,x';\theta)),\quad\Rightarrow g(x,x';\theta)=\mathcal{L}_x\mathcal{L}_{x'}K(x,x';\theta)$

数据

$$\{x_u,y_u\},\{x_f,y_f\},\quad \text{i.e., } x_u=\begin{bmatrix}0\\0\\1\\1\end{bmatrix},\quad y_u=\begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{11.27}$$

$$y_u=u(x)+\epsilon_u,\quad y_f=f(x)+\epsilon_f,\quad \epsilon_f\sim\mathcal{N}(0,\sigma_f^2I)$$

训练

$$\begin{bmatrix}y_u\\y_{ux}\\y_{uxx}\\y_{uxxx}\\y_f\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}K_{uu}&K_{uu'}&K_{uu''}&K_{uu'''}&K_{uf}\\K_{u'u}&K_{u'u'}&K_{u'u''}&K_{u'u'''}&K_{u'f}\\K_{u''u}&K_{u''u'}&K_{u''u''}&K_{u''u'''}&K_{u''f}\\K_{u'''u}&K_{u'''u'}&K_{u'''u''}&K_{u'''u'''}&K_{u'''f}\\K_{fu}&K_{fu'}&K_{fu''}&K_{fu'''}&K_{ff}\end{bmatrix}\right) \tag{11.28}$$

其中,

$$K_{uu}=K(x_u^0,x_u^0;\theta) \tag{11.29}$$

$$K_{uu'}=\frac{\partial}{\partial x}K(x_u^1,x_u^1;\theta)\quad K_{uu''}=\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial x'^2}K(x_u^2,x_u^2;\theta)$$

$$K_{uu''}=\frac{\partial}{\partial x'}K(x_u^0,x_u^1;\theta)\quad K_{uu''}=\frac{\partial^2}{\partial x\partial x'^2}K(x_u^1,x_u^2;\theta)\quad K_{uu'''}=\frac{\partial^3}{\partial x'^3}K(x_u^2,x_u^3;\theta) \tag{11.30}$$

$$K_{uu''}=\frac{\partial^2}{\partial x'^2}K(x_u^0,x_u^2;\theta)\quad K_{uu'''f}=\frac{\partial}{\partial x}\mathcal{L}_{x'}K(x_u^1,x_f;\theta) \tag{11.31}$$

$$K_{uu'''}=\frac{\partial^3}{\partial x^3}K(x_u^0,x_u^3;\theta)\quad K_{uf}=\mathcal{L}_xK(x_u^0,x_f;\theta) \tag{11.32}$$

$$K_{u'''u'''f}=\frac{\partial^3}{\partial x^3}\frac{\partial^3}{\partial x'^3}K(x_u^3,x_f^3;\theta)\quad K_{ff}=\mathcal{L}_x\mathcal{L}_{x'}K(x_f,x_f;\theta)+\sigma_n^2I \tag{11.33}$$

$$\Rightarrow-\log p(y\mid X)=\frac{1}{2}\log|K|+\frac{1}{2}y^TK^{-1}y+\frac{n_u+n_f}{2}\log 2\pi \tag{11.34}$$

3 数值高斯过程

非线性方程与高斯过程并不兼容...

$$\mathcal{N}_x u(x) = f(x), \quad u(x) \sim \mathcal{GP}(0, K(x, x'; \theta)), \quad \text{那么} f(x) \text{ 不是高斯过程...} \tag{11.35}$$

关键思路：在时间上进行线性化！
例如，Burger 方程：

$$u_t + uu_x + x = \nu u_{xx} \Rightarrow u_t = -uu_x + \nu u_{xx}, \quad \text{具有初始和边界条件} \quad \begin{cases} u(x,0) = \sin(\pi x) \\ u(0,t) = 0 \\ u(1,t) = 0 \end{cases}$$

选择一个时间离散化方案；例如，向后欧拉法：

$$\frac{u^n - u^{n-1}}{\Delta t} = -u^n u_x^n + \nu u_{xx}^n \Rightarrow u^n = n^{n-1} - \Delta t u^n u_x^n + \nu \Delta t u_{xx}^n \tag{11.37}$$

用前一步的后验均值进行近似 μ^{n-1}

$$\Rightarrow u^{n-1} = u^n + \Delta t \mu^{n-1} u_x^n - \nu \Delta t u_{xx}^n \tag{11.38}$$

$$\Rightarrow \mathcal{L}_x u^n = u^{n-1} \Rightarrow \text{这就是我们喜欢的形式！} \tag{11.40}$$

然后

$$u^n(x) \sim \mathcal{GP}(0, K(x, x'; \theta)) \Rightarrow u^{n-1}(x) \sim \mathcal{GP}(0, \mathcal{L}_x \mathcal{L}_{x'} K(x, x'; \theta)) \tag{11.41}$$

我们可以写出似然：

$$\begin{bmatrix} u^n \\ u^{n-1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{u,u}^{n,n} & K_{u,u}^{n,n-1} \\ K_{u,u}^{n,n-1} & K_{u,u}^{n-1,n-1} \end{bmatrix}\right) \tag{11.42}$$

其中

$$K_{u,u}^{n,n} = K(x_u^n, x_u^n; \theta) \tag{11.43}$$

$$K_{u,u}^{n,n-1} = \mathcal{L}_{x'} K(x_u, x_u^{n-1}; \theta) \tag{11.44}$$

$$K_{u,u}^{n-1,n-1} = \mathcal{L}_x \mathcal{L}_{x'} K(x_u^{n-1}, x_u^{n-1}; \theta) + \sigma_{nu}^2 I \tag{11.45}$$

预测

$$\begin{bmatrix} u^n(x^*) \\ u^n \\ u^{n-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{u,u}^{n,n}(x^*, x^*) & K_{u,u}^{n,n}(x^*, x_u^n) & \mathcal{L}_x K(x^*, x_u^{n-1}) \\ K_{u,u}^{n,n}(x_u^n, x^*) & K & \mathcal{L}_x K(x^*, x_u^{n-1}) \\ \mathcal{L}_{x'} K(x_u^{n-1}, x^*) & \mathcal{L}_{x'} K(x_u^{n-1}, x_u^{n-1}) & \end{bmatrix}\right) \tag{11.46}$$

$$\Rightarrow p(u^n(x^*) | u^n, u^{n-1}, x^*) = \mathcal{N}([K_{u,u}^{n,n}(x^*, x_u)\mathcal{L}_x K(x^*, x^{n-1})] K^{-1} \begin{bmatrix} u^n \\ u^{n-1} \end{bmatrix} \tag{11.47}$$

$$K_{u,u}^{n,n}(x^*, x^*) - K(x^*, X)K^{-1}K(X, x^*) + K(x^*, X)K^{-1} \begin{bmatrix} 0 & 0 \\ 0 & \Sigma^{n-1,n_1} \end{bmatrix} K^{-1} K(X, x^*) \tag{11.48}$$

其中 $\mu^n = \left[K_{u,u}^{n,n}(x^*, x_{u_n})\mathcal{L}_x K(x^*, x_{u_{n-1}})\right] K^{-1} \begin{bmatrix} u^n \\ u^{n-1} \end{bmatrix}$ and

$$\Sigma^{n,n}(x^*, x^*) = K_{u,u}^{n,n}(x^*, x^*) - K(x^*, X)K^{-1}K(X, x^*) + K(x^*, X)K^{-1} \begin{bmatrix} 0 & 0 \\ 0 & \Sigma^{n-1,n_1} \end{bmatrix} K^{-1} K(X, x^*).$$

Chapter 12

无监督学习：主成分分析、高斯过程潜变量模型

- 霍特林变换（Hotelling transformation）。
- 卡尔霍宁-洛埃夫分解（Karhunen-Loeve decomposition）。
- 适当的正交分解（Proper orthogonal decomposition）。
- 奇异值分解（Singular value decomposition）。

1 主成分分析

设置 Setup： 给定数据 $(x_1, \ldots, x_n), x_i \in \mathbb{R}^d$.
目标： 将数据编码为低维表示：$z_i = f(x_i), z_i \in \mathbb{R}^q, q \ll d$. 即将数据投影到低维子空间上。

最大方差形式（Hotelling 1933）

首先，考虑一维子空间 (sub-space)，即 $q = 1$.
我们可以定义该子空间的坐标 $u_1 \in \mathbb{R}^d$ 使其单位范数 $\|u_1\| = u_1^\top u_1 = 1$.
每个数据点 x_n 被投影到该维度上，得到标量 $u_1^\top x_n$.
因此，所有投影数据的均值为 $u_1^\top \bar{x}$，其中：

$$\bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \text{是样本均值} \tag{12.1}$$

投影数据的方差为：$\frac{1}{n}\sum_{i=1}^{n}(u_1^\top x_n - u_1^\top \bar{x})^2 = u_1^\top S u_1$，其中 S 是样本协方差矩阵：

$$S := \frac{1}{n}\sum_{i=1}^{n}(x_n - \bar{x})(x_n - \bar{x})^\top. \tag{12.2}$$

我们寻找能够捕捉数据中最大方差的方向 u_1，即（“最佳” 总结数据）。

$$u_1^* = \arg\max_{u_1}\{u_1^\top S u_1 + \lambda_1(1 - u_1^\top u_1)\} \tag{12.3}$$

首先分别对 z_{ni} 最小化，得到 $z_{nj} = x_n^\top u_j$（将导数设为零并利用正交性），然后对 b_i 最小化，得到 $b_j = \bar{x}^\top u_j, j = q+1, \ldots, d$.
使用 $x_n = \sum_{i=1}^{d}(x_n^\top u_i)u_i$，我们有：

$$x_n - \tilde{x}_n = \sum_{i=q+1}^{d}\left((x_n - \bar{x})^\top u_i\right)u_i \tag{12.12}$$

因此，差异向量 $x_n - \tilde{x}_n$ " 存在于" 正交于以下空间：

$$J = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=q+1}^{d}\left(x_i^\top u_j - \bar{x}^\top u_j\right)^2 = \sum_{j=q+1}^{d} u_j^\top S u_j \tag{12.13}$$

首先，我们需要通过最小化损失来识别 u_j。与之前一样，这将导致：$Su_i = \lambda_i u_i$，即主成分对应于样本协方差矩阵的特征向量。

给定最优的 $\{u_i\}$，我们还可以看到：$J = \sum_{i=q+1}^{d} \lambda_j$。

实际实现

回顾 SVD：

$$M = U\Sigma V^\top, \quad M \in \mathbb{R}^{n\times d}, \quad U \in \mathbb{R}^{n\times n}, \quad \Sigma \in \mathbb{R}^{n\times d}, \quad W \in \mathbb{R}^{d\times d} \tag{12.14}$$

其中

- U 是正交矩阵
- Σ 是对角矩阵
- $V^\top$ 是正交矩阵

给定数据矩阵 $X \in \mathbb{R}^{n\times d}$，我们首先将其归一化为零均值，即 $\mathbb{E}[X] = 0$。
然后构造无偏样本协方差矩阵：$S = \frac{1}{n}X^\top X$。
计算 S 的特征值分解或 SVD：

$$S = W\Lambda W^\top \tag{12.15}$$

选择潜在空间的维度 (q)，并相应地保留最大的 q 个特征值和特征向量。
然后编码为：

$$Z = XW, \quad Z \in \mathbb{R}^{n\times q}, \quad X \in \mathbb{R}^{n\times d}, \quad W \in \mathbb{R}^{d\times q} \tag{12.16}$$

我们还可以从编码值重建 X：

$$X = ZW^\top, \quad X \in \mathbb{R}^{n\times d}, \quad Z \in \mathbb{R}^{n\times q}, \quad W \in \mathbb{R}^{d\times q} \tag{12.17}$$

2 概率主成分分析

我们可以在概率背景下公式化 PCA 为：

$$p(z) = \mathcal{N}(z|0, I), \quad p(x|z) = \mathcal{N}(x|zW^\top + \mu, \sigma^2 I) \tag{12.18}$$

并使用 MLE 来确定未知的模型参数：$\theta := \{W, \mu, \sigma^2\}$。为了做到这一点，我们需要边际似然的表达式：

$$p(x) = \int p(x|z)p(z)\,dz \tag{12.19}$$

因为这是一个线性-高斯模型，所以它有一个闭合形式的解：

$$p(x) = \mathcal{N}(x|\mu, C), \quad \text{其中} C = WW^\top + \sigma^2 I \tag{12.20}$$

因为

$$\mathbb{E}[x] = \mathbb{E}[zW^\top + \mu + \epsilon] = \mu \tag{12.21}$$

$$\text{cov}[x] = \mathbb{E}[(zW^\top + \mu + \epsilon)(zW^\top + \mu + \epsilon)^\top] = \mathbb{E}[Wzz^\top W^\top] + \mathbb{E}[\epsilon\epsilon^\top] = WW^\top + \sigma^2 I \tag{12.22}$$

此外，我们可以获得潜在变量的后验：

$$p(z|x) = \mathcal{N}(z|M^{-1}(X-\mu)W, \sigma^2 M^{-1}), \quad \text{其中} M = W^\top W + \sigma^2 I \tag{12.23}$$

为了估计模型参数，我们有：

$$-\log p(X|\mu, W, \sigma^2) = -\sum_{i=1}^{n} \log p(x_i|W, \mu, \sigma^2) \tag{12.24}$$

$$= \frac{nd}{2}\log 2\pi + \frac{n}{2}\log|C| + \frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^\top C^{-1}(x_i-\mu) \tag{12.25}$$

然后我们可以得到：

$$\mu_{ML} = \bar{x}, \Rightarrow -\log p(X|W, \mu, \sigma^2) = \frac{n}{2}\left\{d\log(2\pi) + \log|C| + \text{Tr}(C^{-1}S)\right\} \tag{12.26}$$

其中 S 是样本协方差。最大化这个关于 W 和 σ^2 的表达式，我们还可以得到：

$$W_{ML} = U_m(L_m - \sigma^2 I)^{1/2}R, \tag{12.27}$$

其中 U_m 是一个 $d \times q$ 矩阵，其列是 S 的特征向量。L_m 是包含 S 特征值的对角矩阵，而 R 是一个任意的正交矩阵。

$$\sigma^2_{ML} = \frac{1}{d-q}\sum_{i=q+1}^{d}\lambda_i \tag{12.28}$$

3 行业技巧 Tricks of the Trade

(1) 变分界限

我们有一些损失函数 $\mathcal{L}(\theta)$，它在计算上是不可处理的（通常涉及一些不可处理的边际化）。我们无法评估它，更不用说最小化它了！例如，在潜变量模型中：

$$p(x|z) = \frac{p_\theta(x|z)p(z)}{p_\theta(x)} \tag{12.28}$$

$$p_\theta(x) = \int p_\theta(x,z)\,dz = \int p_\theta(x|z)p(z)\,dz \tag{12.29}$$

在 PPCA 中，我们假设 $p_\theta(x|z)$ 和 $p(z)$ 是高斯的，并且 z 对 x 的依赖是线性的，即

$$p_\theta(x|z) = \mathcal{N}(x|zW^\top + \mu, \sigma^2 I), \quad \theta := \{\mu, W, \sigma^2\} \tag{12.30}$$

然后我们可以解析计算

$$p_\theta(x) = \mathcal{N}(x|\mu, C), \quad C := WW^\top + \sigma^2 I \tag{12.31}$$

但在一般情况下并非如此（即 $p_\theta(x|z)$ 和 $p(z)$ 可能是非高斯的，或 x 对 z 的依赖可能是非线性的）。因此 $p_\theta(x)$ 将是不可处理的。

技巧：引入一些辅助参数 φ 和下界：

$$\theta^* = \arg\min_\theta L(\theta) \le \arg\min_{\theta,\varphi}\hat{L}(\theta, \varphi) \tag{12.32}$$

需要回忆的想法： - **詹森不等式**：$f(\mathbb{E}_{x\sim p(x)}[x]) \leq \mathbb{E}_{x\sim p(x)}[f(x)]$，如果 f 是一个凸函数。- **重要性采样**：$\mathbb{E}_{x\sim p(x)}[f(x)] = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{p(x_i)}{q(x_i)}, x_i \sim q(x)$。

公式化： 我们通常想要最小化：

$$-\log p_\theta(x) = -\log \int p_\theta(x,z)\,dz = -\log \int \frac{p_\theta(x,z)}{q_\varphi(z|x)} q_\varphi(z|x)\,dz \tag{12.33}$$

$$= -\log \mathbb{E}_{z\sim q_\varphi(z|x)}\left[\frac{p_\theta(x,z)}{q_\varphi(z|x)}\right] \leq -\mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\varphi(z|x)}\right] \tag{12.34}$$

$$= -\int \log \frac{p_\theta(x,z)p(z)}{q_\varphi(z|x)} q_\varphi(z|x)\,dz = -\int \log p_\theta(x|z) q_\varphi(z|x)\,dz - \int \log \frac{p(z)}{q_\varphi(z|x)} q_\varphi(z|x)\,dz \tag{12.35}$$

$$= -\mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] + \mathrm{KL}(q_\varphi(z|x)||p(z)) \geq -\log p(x) \tag{12.36}$$

我们注意到 $\hat{L}(\theta,\varphi) = -\mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] + \mathrm{KL}(q_\varphi(z|x)||p(z))$。

假设对 $q_\varphi(z|x)$ 和 $p_\theta(x|z)$ 使用参数模型，这个表达式是可计算的，并且可以针对 $\{\theta,\varphi\}$ 进行优化。

注意 ：为了优化这个目标，我们需要一个算法。直接的蒙特卡洛估计是：

$$\nabla_\varphi \mathbb{E}_{z\sim q_\varphi(z|x)}[\log p_\theta(x|z)] = \mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\nabla_\varphi \log q_\varphi(z|x)\right] \approx \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x|z_i)\nabla_\varphi \log q_\varphi(z_i|x_i) \tag{12.37}$$

我们使用了：

$$\nabla_\varphi \log q_\varphi(z|x) = \frac{\nabla_\varphi q_\varphi(z|x)}{q_\varphi(z|x)} \tag{12.38}$$

不幸的是，这个梯度估计会表现出非常高的方差，实际上是无用的。这导致我们进入下一个技巧...

(2) 重参数化技巧

如果我们能找到一个函数 $h:(\epsilon,\varphi)\to z$，它对 φ 是可微的，并且有一个概率分布 $p(\epsilon)$ 定义在 ϵ 上，这个分布易于抽样，并且满足：

$$q_\varphi(z|x) = h_\varphi(x,\epsilon), \quad \epsilon\sim p(\epsilon) \quad \text{生成样本} \quad z\sim q_\varphi(z|x) \tag{12.39}$$

那么我们可以估计所需的梯度：

$$\nabla_\varphi \mathbb{E}_{z\sim q_\varphi(z|x)}[\log p_\theta(x|z)] = \mathbb{E}_{\epsilon\sim p(\epsilon)}[\nabla_\varphi \log p_\theta(x|h_\varphi(x,\epsilon))] \tag{12.40}$$

现在梯度与我们进行期望的分布无关！

推导： 我们想证明：

$$\nabla_\varphi \mathbb{E}_{z\sim q_\varphi(z|x)}[f(z)] = \mathbb{E}_{\epsilon\sim p(\epsilon)}[\nabla_\varphi f(h(\epsilon,\varphi))] \tag{12.41}$$

使用变量变换公式变换 $p(\epsilon)$：

$$p(z) = \left|\frac{d\epsilon}{dz}\right| p(\epsilon) = |p(\epsilon)| \tag{12.42}$$

然后我们重新表达与重新参数化密度相关的麻烦期望：

$$\nabla_\varphi \mathbb{E}_{z\sim q_\varphi(z|x)}[f(z)] = \nabla_\varphi \int f(z)q_\varphi(z)\,dz = \nabla_\varphi \int f(z)p(\epsilon)\,d\epsilon \tag{12.43}$$

$$= \nabla_\varphi \int f(h(\epsilon,\varphi))p(\epsilon)\,d\epsilon = \mathbb{E}_{\epsilon\sim p(\epsilon)}[\nabla_\varphi f(h(\epsilon,\varphi))] \tag{12.44}$$

这返回了梯度的无偏蒙特卡洛估计！

(3) 密度比技巧 密度比在统计学和机器学习中是无处不在的。当我们比较两个密度时，它们会持续出现：

$$r(x) = \frac{p(x)}{q(x)} \tag{12.46}$$

例如，贝叶斯规则、KL 散度、重要性采样等...

在实际中，估计密度比往往非常困难，因为 $p(x)$ 和 $q(x)$ 可能是不可处理的高维分布，我们可能只拥有它们的样本，但不知道它们的闭合形式表达。假设我们可以创建一个包含 $2N$ 点的数据集：- 从 $p(x)$ 中抽取 N 个数据点并标记为 $+1$。- 从 $q(x)$ 中抽取 N 个数据点并标记为 -1。

通过这种构造，我们可以将这些概率写成条件形式：

$$p(x) = p(x|y=+1), \quad q(x) = p(x|y=-1), \quad \text{贝叶斯：} p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{12.47}$$

然后：

$$r(x) = \frac{p(x)}{q(x)} = \frac{p(x|y=+1)}{p(x|y=-1)} = \frac{\frac{p(y=+1|x)p(x)}{p(y=+1)}}{\frac{p(y=-1|x)p(x)}{p(y=-1)}} \tag{12.48}$$

$$= \frac{p(y=+1|x)}{p(y=-1|x)} = \frac{p(y=+1|x)}{1-p(y=-1|x)} = \frac{S(x)}{1-S(x)}, \tag{12.49}$$

其中 $S(x)$ 是判别器的输出。

由于我们对 $p(x)$ 和 $q(x)$ 有相等数量的样本，因此 $p(y=+1) = p(y=-1) = 0.5$。因此，密度比估计的问题等价于二分类问题！而这正是我们擅长解决的事情:)

Chapter 13

变分自编码器和条件变分自编码器

1 变分自编码器

设置 Setup: 我们给定数据 $\{x_i\}, i = 1, \ldots, n, x_i \in \mathbb{R}^d$（通常是高维的）。**目标 Goal:** 学习生成观察数据的分布 $p(x)$。

为此，我们假设存在一组潜在变量 $z \in \mathbb{R}^m$，它们解释了 x 的变化性（例如，如果 x 是一张图像，z 可能包括光照条件等）。

具体来说，我们假设一个模型 $p_\theta(x|z)$，这样我们可以构造 $p(x)$ 的近似为：

$$p_\theta(x, z) = \int p_\theta(x|z)p(z)dz, \tag{13.1}$$

其中 $p(z)$ 是潜在变量的先验分布。这定义了一种所谓的生成模型：

$$p_\theta(x, z) = p_\theta(x|z)p(z) \tag{13.2}$$

因此，提出这些潜在变量应该是什么是自然的？换句话说，后验分布是什么：

$$p(z|x) = \frac{p_\theta(x|z)p(z)}{p(x)}, \tag{13.3}$$

但这是不可行的... 因此，我们将尝试通过引入一组可计算的量来近似它：

$$p(z|x) \approx q_\varphi(z|x) = \mathcal{N}(\mu_\varphi(x), \Sigma_\varphi(x)) \tag{13.4}$$

$$p_\theta(x|z) = \mathcal{N}(\mu_\theta(x), \Sigma_\theta(x)) \tag{13.5}$$

$$p(z) = \mathcal{N}(0, I) \tag{13.6}$$

这是 Kingma 和 Welling 的经典 VAE 设置。

我们希望 $q_\varphi(z|x)$ 能够很好地近似真实后验分布（即）：

$$\mathrm{KL}[q_\varphi(z|x)\|p(z|x)] := -\int \log \frac{p(z|x)}{q_\varphi(z|x)} q_\varphi(z|x)dz$$

$$= -\int \left[\log p_\theta(x, z) + \log q_\varphi(z|x) - \log p(x) - \log q_\varphi(z|x)\right] q_\varphi(z|x)dz$$

$$= -\int \log p_\theta(x, z) q_\varphi(z|x)dz + \int \log \frac{q_\varphi(z|x)}{p(z)} q_\varphi(z|x)dz$$

$$\Rightarrow -\log p_\theta(x) + \mathrm{KL}[q_\varphi(z|x)\|p(z)] = \mathrm{KL}[q_\varphi(z|x)\|p(z)] - \mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right]$$

$$-\log p(x) \le \mathcal{L}(\theta, \varphi)$$

在这里我们使用了 $\mathrm{KL}[q_\varphi(z|x)\|p(z|x)] \ge 0$ 的现实情况，并注意到 ELBO 为
$\mathcal{L}(\theta, \varphi) = \mathrm{KL}[q_\varphi(z|x)\|p(z)] - \mathbb{E}_{z\sim q_\varphi(z|x)}[\log p_\theta(x|z)]$。

回顾一下，我们需要 $\nabla_\varphi \mathcal{L}, \nabla_\theta \mathcal{L}$，但 $\mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right]$ 是麻烦的... 同时回顾重参数化技巧：

$$\nabla_\varphi \mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] = \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\nabla_\varphi \log p_\theta(x|h_\varphi(x,\epsilon))\right] \tag{13.13}$$

在这种情况下，

$$z \sim q_\varphi(z|x) = \mathcal{N}(\mu_\varphi(x), \Sigma_\varphi(x)), \tag{13.14}$$

因此我们可以使用以下简单的重参数化：

$$\nabla_\varphi \mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] = \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\nabla_\varphi \log p_\theta(x|\mu_\varphi(x) + \Sigma_\varphi^{1/2}(x)\epsilon)\right]$$

即，如果 $\epsilon \sim p(\epsilon) = \mathcal{N}(0, I)$，那么：

$$z = \mu_\varphi(x) + \Sigma_\varphi^{1/2}(x)\epsilon \sim q_\varphi(z|x) = \mathcal{N}(\mu_\varphi(x), \Sigma_\varphi(x)). \tag{13.16}$$

ELBO 计算

$$x_i \xrightarrow{q_\varphi(z|x)} z_i = \mu_\varphi(x) + \epsilon\Sigma_\varphi^{1/2}(x) \tag{13.17}$$

$$z_i \xrightarrow{p_\theta(z|x)} x_i \tag{13.18}$$

$$\mathrm{KL}[q_\varphi(z|x)\|p(z)] = \frac{1}{2}\left[\mu_\varphi(x)^T\mu_\varphi(x) - m - \log|\Sigma_\varphi(x)| + \mathrm{tr}\{\Sigma_\varphi(x)\}\right] \tag{13.19}$$

$$\mathbb{E}_{z\sim q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] \approx \frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x_i|z_i), \quad \text{where } z_i = \mu_\varphi(x) + \epsilon\Sigma_\varphi^{1/2}(x),\ \epsilon \sim \mathcal{N}(0, I). \tag{13.20}$$

$$= \frac{1}{2}\log|\Sigma_\varphi(x)| + \frac{1}{2}(x - \mu_\theta(x))^T\Sigma_\theta^{-1}(x)(x - \mu_\theta(x)) + \frac{nd}{2}\log 2\pi \tag{13.21}$$

2 条件变分自编码器

给定数据 $\{x_i, y_i\}, x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

其中我们应该注意到 $p(x, y, z) = p(y|x, z)p(z|x)p(x)$。

再次问这些潜在变量应该是什么，即后验是什么？

$$p(z|x, y) = \frac{p_\theta(y|x, z)p_\gamma(z|x)}{p_\theta(y|x)}. \tag{13.23}$$

对于一般选择的 $p_\theta(y|x, z)$ 和 $p_\gamma(z|x)$，这个后验是不可解的，我们将引入一个变分近似：

$$p(z|x, y) \approx q_\varphi(z|x, y). \tag{13.24}$$

我们希望这个近似能够最小化 KL 散度：

$$\mathrm{KL}[q_\varphi(z|x,y)\|p(z|x,y)] = -\int \log \frac{p(z|x,y)}{q_\varphi(z|x,y)} q_\varphi(z|x,y)dz \tag{13.25}$$

$$= -\int \left[\log p_\theta(y|x,z) + \log p_\gamma(z|x) - \log p_\theta(y|x) - \log q_\varphi(z|x,y)\right] q_\varphi(z|x,y)dz \tag{13.26}$$

$$= -\int \log p_\theta(y|x,z) q_\varphi(z|x,y)dz + \int \log \frac{p_\gamma(z|x)}{q_\varphi(z|x,y)} q_\varphi(z|x,y)dz \tag{13.27}$$

$$\Rightarrow \log p_\theta(y|x) + \mathrm{KL}[q_\varphi(z|x,y)\|p_\gamma(z|x)] = \mathrm{KL}[q_\varphi(z|x,y)\|p_\gamma(z|x)] - \mathbb{E}_{z\sim q_\varphi(z|x,y)}[\log p_\theta(y|x,z)]. \tag{13.28}$$

ELBO 可以直接计算，如果我们引入以下表示：

$$p_\theta(y|x,z) = \mathcal{N}(\mu_\theta(x,z), \Sigma_\theta(x,z)) \tag{13.29}$$

$$q_\varphi(z|x,y) = \mathcal{N}(\mu_\varphi(x,y), \Sigma_\varphi(x,y)) \tag{13.30}$$

$$p_\gamma(z|x) = \mathcal{N}(\mu_\gamma(x), \Sigma_\gamma(x)) \tag{13.31}$$

工作流程如下：

$$(x,y) \xrightarrow{q_\varphi(z|x,y)} \mu_\varphi(x,y), \Sigma_\varphi(x,y) \tag{13.32}$$

$$x \xrightarrow{p_\gamma(z|x)} \mu_\gamma(x), \Sigma_\gamma(x) \tag{13.33}$$

$$z = \mu_\varphi(x) + \epsilon\Sigma_\varphi^{1/2}(x,y), \quad \epsilon \sim \mathcal{N}(0,I). \tag{13.34}$$

$$(x,z) \xrightarrow{p_\theta(y|x,z)} \mu_\theta(x,z), \Sigma_\theta(x,z) \tag{13.35}$$

Chapter 14

生成对抗网络

1 设置

给定数据 $\{x_i\}$, $i = 1, \ldots, n$, $x_i \in \mathbb{R}^d$，我们的目标是学习真实的数据生成分布。

- $p(x)$: 真实数据分布。
- $p_\theta(x)$: 模型分布。
- $q(x)$: 经验数据分布。

到目前为止，我们构建了概率模型，并使用最大似然估计对它们进行了训练。例如，在变分自编码器（VAEs）中，我们考虑了 $p(x)$ 的生成模型：

$$p_\theta(x) = \int p(x, z)dz = \int p(x|z)p(z)dz, \quad z \sim p(z), \quad z \in \mathbb{R}^q, \tag{14.1}$$

其中 $p_\theta(x)$ 是边际模型分布。我们的目标是使其尽可能接近真实数据分布。为此，我们引入了适当的近似，并构建了可计算的边际似然下界：

$$\mathcal{L}(\theta, \varphi) := -\log p_\theta(x) \leq \mathrm{KL}[q_\varphi(z|x)\|p(z)] - \mathbb{E}_{z\sim q_\varphi(z|x)}[\log p_\theta(x|z)] \tag{14.2}$$

（为避免混淆，$q_\varphi(z|x)$ 不应与经验分布混淆）。给定从经验数据分布中抽取的数据 $\{x_i\}$，即 $x \sim q(x)$，我们评估并优化此目标以获得 $\{\theta^*, \varphi^*\}$。

这种最大似然估计方法可以从理论上证明等价于最小化经验数据分布与模型边际分布之间的 KL 散度：

$$\{\theta^*, \varphi^*\} = \arg\min_{\theta,\varphi} \mathrm{KL}[q(x)\|p_\theta(x)] \tag{14.3}$$

2 生成对抗网络（GANs）

在这里，我们引入一个生成模型：

$G_\theta(z), z \sim p(z)$, $G_\theta(\cdot)$ 是一个由θ 参数化的确定性映射，若$x = G_\theta(z)$, 则$x \sim p_\theta(x) \approx p(x)$, $G : \mathbb{R}^q \to \mathbb{R}^d$.

与最大似然估计方法不同，这里我们引入一个判别器 $D_\varphi(x), D : \mathbb{R}^d \to [0, 1]$，它表示 x 是来自经验数据分布 $q(x)$ 的概率，而不是来自模型分布 $p_\theta(x)$（由生成器产生的 x'）。

然后，我们训练判别器 $D_\varphi(x)$ 为来自 $q(x)$ 的样本和来自 $p_\theta(x)$ 的样本分配正确的标签。我们同时训练生成器 $G_\theta(z)$ 来最小化 $\log(1 - D_\varphi(G_\theta(z)))$，即为了欺骗判别器，使其认为生成的样本来自数据分布。

3 理论结果（在非参数极限下）

(1)

对于固定的 $G_\theta(z)$，最优判别器为：

$$D_\varphi^*(x) = \frac{q(x)}{q(x) + p_\theta(x)}. \tag{14.4}$$

判别器 $D_\varphi(x)$ 的训练目标可以重写为：

$$c(\theta) = \max_\varphi \mathcal{L}(\theta, \varphi) = \mathbb{E}_{x\sim q(x)}\left[\log \frac{q(x)}{q(x) + p_\theta(x)}\right] + \mathbb{E}_{x\sim p_\theta(x)}\left[\log \frac{p_\theta(x)}{q(x) + p_\theta(x)}\right]. \tag{14.5}$$

该目标的全局最小值当且仅当 $p_\theta(x) = q(x)$ 时达到。在那一点上，$c(\theta) = -\log 4$。这个目标也可以重写为：

$$c(\theta) = -\log 4 + \mathrm{KL}\left[q(x)\middle\|\frac{q(x) + p_\theta(x)}{2}\right] + \mathrm{KL}\left[p_\theta(x)\middle\|\frac{q(x) + p_\theta(x)}{2}\right] \tag{14.6}$$

$$\Rightarrow c(\theta) = -\log 4 + 2\,\mathrm{JSD}(q(x)\|p_\theta(x)) \tag{14.7}$$

注意到 $\mathrm{JSD}(q(x)\|p_\theta(x)) = \mathrm{KL}\left[q(x)\middle\|\frac{q(x)+p_\theta(x)}{2}\right] + \mathrm{KL}\left[p_\theta(x)\middle\|\frac{q(x)+p_\theta(x)}{2}\right]$ 是 $p_\theta(x)$ 和 $q(x)$ 之间的 Jensen-Shannon 散度。与 KL 散度不同，这是一种对称散度。

(2)

如果 $G_\theta(z)$ 和 $D_\varphi(x)$ 具有足够的能力，并且在算法的每一步中，判别器允许根据 $G_\theta(z)$ 达到其最优解，同时更新 $p_\theta(x)$ 以改善标准：

$$\mathbb{E}_{x\sim q(x)}\left[\log D_\varphi^*(x)\right] + \mathbb{E}_{x\sim p_\theta(x)}\left[\log(1 - D_\varphi^*(x))\right] \tag{14.8}$$

那么 $p_\theta(x)$ 会收敛到 $q(x)$。这个过程定义了两个参与者之间的零和最小-最大博弈，结果是一个目标函数：

$$\min_\theta \max_\varphi \mathbb{E}_{x\sim q(x)}\left[\log D_\varphi(x)\right] + \mathbb{E}_{x\sim p_\theta(x)}\left[\log(1 - D_\varphi(G_\theta(z)))\right] := \mathcal{L}(\theta, \varphi) \tag{14.9}$$

在实践中，这可能不会为 $G_\theta(\cdot)$ 提供足够的梯度来进行训练。最开始时，$G_\theta(z)$ 提供的样本质量较差，$D_\varphi(x)$ 以高置信度拒绝它们，因为它们显然与训练数据不同，从而使得 $\log(1 - D_\varphi(G_\theta(z)))$ 饱和。

相反，我们可以训练 $G_\theta(z)$ 来最大化 $\log D_\varphi(G_\theta(z))$，即：

$$\max_\theta \min_\varphi \mathbb{E}_{x\sim q(x)}\left[\log D_\varphi(x)\right] + \mathbb{E}_{z\sim p(z)}\left[\log D_\varphi(G_\theta(z))\right] \tag{14.10}$$

4 GAN 算法

归一化零均值和单位方差的数据对 $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_n\}$，$\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_n\}$，每次迭代中判别器和生成器的更新相对数量 N_d 和 N_g。

Algorithm 1 GAN Algorithm

while iter $< T$ **do**
  $k \leftarrow 0$ ▷ Initialize inner loop counter
  while $k < t$ **do**
    Sample mini-batch of m noise samples $\{z_1, \ldots, z_m\}$ from prior distribution $p(z)$
    Sample mini-batch of m data points $\{x_1, \ldots, x_m\}$
    $\varphi_{n+1} \leftarrow \varphi_n + \eta \nabla_\varphi \left(\frac{1}{m} \sum_{i=1}^m [\log D_\varphi(x_i) + \log(1 - D_\varphi(G_\theta(z_i)))] \right)$
    where θ is kept fixed
    $k \leftarrow k + 1$
  end while
  Sample mini-batch of m noise samples $\{z_1, \ldots, z_m\}$ from $p(z)$
  $\theta_{n+1} \leftarrow \theta_n + \eta \nabla_\theta \left(\frac{1}{m} \sum_{i=1}^m [\log(1 - D_\varphi(G_\theta(z_i)))] \right)$
  iter $\leftarrow$ iter $+ 1$
end while

5 对抗学习推断

回顾在 GAN 中我们引入了一个生成模型：

$$x = G_\theta(z), \quad z \sim p(z), \quad \text{目标是} x \sim p_\theta(x) \approx q(x). \tag{14.11}$$

其中 $G_\theta : \mathbb{R}^q \to \mathbb{R}^d$ 是一个确定性的生成器/解码器。为了有效地训练这个模型，我们引入了一个判别器 $D_\varphi : \mathbb{R}^d \to [0, 1]$，它学习区分来自 $p_0(x)$ 的样本和来自 $q(x)$ 的样本，从而导致一个对抗优化目标：

$$\min_\theta \max_\varphi \mathbb{E}_{x \sim q(x)}[\log D_\varphi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\varphi(G_\theta(z)))] := \mathcal{L}(\theta, \varphi). \tag{14.12}$$

总结 GAN 的设置旨在匹配：

- 经验数据分布 $q(x)$。
- 模型边际分布 $p_\theta(x) = \int p_\theta(x|z)p(z)\, dz$。

通过最小化（理论上）$\mathrm{JSD}(q(x) \| p_\theta(x))$。然而，这个模型无法对潜在变量 z 进行推断（即，我们不知道数据中的那些变量应该是什么，我们不知道后验 $p(z|x)$）。

为此，我们可以引入一个新的模型，该模型能够对 z 进行推断。为此，我们将考虑联合分布：

- 编码器联合：$q_\varphi(x, z) = q_\varphi(z|x)q(x)$
- 解码器联合：$p_\theta(x, z) = p_\theta(x|z)p(z)$

为了以对抗的方式学习这个模型，我们再次需要引入一个判别器 $T_\psi(x, z)$，它学习如何区分从 $q_\varphi(x, z) = q_\varphi(z|x)q(x)$ 中抽取的样本和从 $p_\theta(x, z) = p_\theta(x|z)p(z)$ 中抽取的样本。与之前一样，判别器被训练以准确学习如何区分样本，而生成器被训练以生成能够欺骗判别器的联合样本。这一次，我们还可以学习编码器 $q_\varphi(z|x)$。

$$\min_{\theta, \varphi} \max_\psi \mathbb{E}_{x \sim q(x)} \left[\log T_\psi(x, z)\right] + \mathbb{E}_{z \sim p(z)} \left[\log(1 - T_\psi(x, z))\right]. \tag{14.13}$$

同样，对于固定的 $p_\theta(x|z)$ 和 $q_\varphi(z|x)$，最优判别器为：

$$T^*_\psi(x, z) = \frac{q_\varphi(x, z)}{q_\varphi(x, z) + p_\theta(x, z)}. \tag{14.14}$$

给定这个最优判别器，上述对抗游戏最小化了 $p_\theta(x,z)$ 和 $q_\varphi(x,z)$ 之间的 Jensen-Shannon 散度。匹配联合分布还会导致匹配边际分布，即 $p_\theta(x) \approx q(x)$，以及条件分布 $p(z|x) \approx q_\varphi(z|x)$ 和 $p_\theta(x|z) \approx q(x|z)$。

与 GAN 类似，为了减轻梯度消失问题，在实践中，我们优化：

$$\max_{\theta,\psi} \min_{\theta,\psi} \mathbb{E}_{x\sim q(x)} \mathbb{E}_{z\sim q(z)} \left[\log T_\psi(x,z)\right]. \tag{14.15}$$

使用重参数化技巧评估梯度时需要特别小心，即, $p_\theta(x|z) = f_\theta(z,\epsilon)$, $q_\varphi(z|x) = f_\varphi(x,\epsilon)$, $\epsilon \sim \mathcal{N}(0,I)$.