Statistical Computing
Chap. 1: Solving Nonlinear Equations

LIU, Ran

Department of Statistics,
Beijing Normal University (Zhuhai Campus)

February 26, 2024

北京師範大學
BEIJING NORMAL UNIVERSITY

Summary

Likelihood

给定输出 $\boldsymbol{X}$ 时，关于参数 $\boldsymbol{\Theta}$ 的似然函数 $L(\boldsymbol{\Theta} \mid \boldsymbol{X})$（在数值上）等于给定参数 $\boldsymbol{\Theta}$ 后变量 $\boldsymbol{X}$ 的概率。

如果 $\boldsymbol{X} = \{x_1, \ldots, x_n\}$ 是独立同分布（i.i.d.）的，每个变量都服从密度函数 $f(x \mid \boldsymbol{\Theta})$，该密度函数依赖于 p 个未知参数组成的向量 $\boldsymbol{\Theta} = \{\theta_1, \ldots, \theta_p\}$，那么联合似然函数为

$$L(\boldsymbol{\Theta} \mid \boldsymbol{X}) = \prod_{i=1}^{n} f(x_i \mid \boldsymbol{\Theta}).$$

当数据不是独立同分布时，联合似然函数仍然表示为关于 $\boldsymbol{\Theta}$ 的联合密度 $f(x_1, \ldots, x_n \mid \boldsymbol{\Theta})$。

Example

假设我们在抛硬币，总共抛 6 次，结果为（正，反，正，正，反，反），那么设正面概率为 p，此时的似然函数为

$$L(\boldsymbol{\Theta} \mid \boldsymbol{X}) = p^3(1-p)^3.$$

它在数值上等于出现结果为（正，反，正，正，反，反）的概率.

注意，我们这里事件是 6 维的变量 $\boldsymbol{X} = (x_1, x_2, x_3, x_4, x_5, x_6)$，更加规范的书写应为

$$L(\boldsymbol{\Theta} \mid \boldsymbol{X}) = p^{\sum_{i=1}^{6} I(x_i=1)}(1-p)^{\sum_{i=1}^{6} I(x_i=0)},$$

若假设观测到的数据是正面的次数 m，也就是 $\boldsymbol{X} = m$，似然函数为

$$L(\boldsymbol{\Theta} \mid \boldsymbol{X}) = C_6^m p^m (1-p)^{(6-m)}.$$

Example

似然函数并不只针对一个分布的观测值，而是所有的观测值，无论他们的分布。

假设 $z_1, z_2, ..., z_n$（独立同分布）来自正态分布 $N(\mu, \sigma^2)$，并且 $y_1, y_2, ..., y_m$（独立同分布）来自泊松分布 $Pois(\lambda)$，则定义 $\boldsymbol{X} = \{z_1, z_2, ..., z_n, y_1, y_2, ..., y_m\}$ 和 $\boldsymbol{\Theta} = \{\mu, \sigma, \lambda\}$，那么

$$
\begin{aligned}
L(\boldsymbol{\Theta} \mid \boldsymbol{X}) &= \prod_{i=1}^{n} f(z_i \mid \mu, \sigma) \prod_{j=1}^{m} f(y_j \mid \lambda) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp\{-\frac{(z_i - \mu)^2}{2\sigma^2}\} \prod_{j=1}^{m} \frac{\lambda^{y_j} e^{-\lambda}}{y_j!}
\end{aligned}
$$

Summary

Frequentist

对频率学派来说，参数是固定的，通常使用 MLE 的方法来估计参数，也就是寻找以较高概率产生观察数据的参数值。似然函数在 MLE $\hat{\boldsymbol{\Theta}}$ 处达到最大值：

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta} \mid \mathbf{X}) = \arg\max_{\boldsymbol{\Theta}} \ell(\boldsymbol{\Theta} \mid \mathbf{X}),$$

其中 $\ell(\boldsymbol{\Theta} \mid \mathbf{X})$ 是对数似然函数。

在相当普遍的条件下，当 $n \to \infty$ 时，$\hat{\boldsymbol{\Theta}}$ 是渐近无偏的。

如何估计 MLE 的方差？(Fisher 信息)

Score Function

求解 MLE，即寻找 $\ell'(\theta|\boldsymbol{x}) = 0$ 的根。

我们将偏导数 $\ell'(\theta|\boldsymbol{x})$ 称为**得分函数**，它关于 x 在 θ 固定下的条件期望为 0：

$$E\left[\frac{\partial}{\partial\theta}\log f(x;\theta) \mid \theta\right] = \int_{\mathbb{R}} \frac{\frac{\partial}{\partial\theta} f(x;\theta)}{f(x;\theta)} f(x;\theta)dx = \frac{\partial}{\partial\theta}\int_{\mathbb{R}} f(x;\theta)dx = 0.$$

上式因为是关于 x 在 θ 固定下的条件期望，所以 x 可被认为是服从 $f(x;\theta)$ 分布的，所以 $\int_{\mathbb{R}} f(x;\theta)dx = 1$.

Fisher Information

Fisher 信息被定义为得分函数关于 x 的方差：

$$\mathcal{I}(\theta) = \mathrm{E}\left[\left(\frac{\partial}{\partial\theta}\log f(x;\theta)\right)^2 \middle| \theta\right] = \int_{\mathbb{R}}\left(\frac{\partial}{\partial\theta}\log f(x;\theta)\right)^2 f(x;\theta)dx$$

如果 $logf(x;\theta)$ 对于 θ 具有两次可微性，并且在一定的正则条件下，Fisher 信息也可以写成：

$$\mathcal{I}(\theta) = -\mathrm{E}\left[\frac{\partial^2}{\partial\theta^2}\log f(x;\theta)\middle|\theta\right],$$

这是因为

$$\begin{aligned}\frac{\partial^2}{\partial\theta^2}\log f(x;\theta) &= \frac{\partial(\frac{\partial \log f(x;\theta)}{\partial\theta})}{\partial\theta} = \frac{\partial(\frac{1}{f(x;\theta)}\frac{\partial f(x;\theta)}{\partial\theta})}{\partial\theta} \\ &= \frac{\frac{\partial^2}{\partial\theta^2}f(x;\theta)}{f(x;\theta)} - \left(\frac{\frac{\partial}{\partial\theta}f(x;\theta)}{f(x;\theta)}\right)^2 \\ &= \frac{\frac{\partial^2}{\partial\theta^2}f(x;\theta)}{f(x;\theta)} - \left(\frac{\partial}{\partial\theta}\log f(x;\theta)\right)^2\end{aligned}$$

和

$$E\left[\left.\frac{\frac{\partial^2}{\partial\theta^2}f(x;\theta)}{f(x;\theta)}\right|\theta\right] = \frac{\partial^2}{\partial\theta^2}\int_{\mathbb{R}} f(x;\theta)dx = 0.$$

Fisher Information (multivariate)

当变量为多维的情况下，让 $\ell''(\boldsymbol{\theta})$ 表示一个 $p \times p$ 的矩阵，其中第 (i, j) 个元素由 $\partial^2 l(\boldsymbol{\theta}) / (\partial\theta_i \partial\theta_j)$ 给出。**Fisher 信息矩阵**被定义为：

$$\mathcal{I}(\boldsymbol{\theta}) := E\left\{\ell'(\boldsymbol{\theta})\ell'(\boldsymbol{\theta})^{\mathrm{T}}\right\} = -E\left\{\ell''(\boldsymbol{\theta})\right\} = -E\left\{\nabla^2 \ell(\boldsymbol{\theta})\right\}$$

其中期望是针对 $\mathbf{X}_1, \ldots, \mathbf{X}_n$ 固定 $\boldsymbol{\theta}$ 的分布求得的。$\mathcal{I}(\boldsymbol{\theta})$ 有时被称为**期望 Fisher 信息**，以区别于 $-\ell''(\boldsymbol{\theta})$，它是**观测 Fisher 信息**。

在正则条件下，MLE $\widehat{\boldsymbol{\theta}}$ 的渐近方差-协方差矩阵是 $\mathcal{I}\left(\boldsymbol{\theta}^*\right)^{-1}$，其中 $\boldsymbol{\theta}^*$ 表示 $\boldsymbol{\theta}$ 的真实值。事实上，当 $n \to \infty$ 时，$\hat{\boldsymbol{\theta}}$ 的极限分布是

$$N_p\left(\boldsymbol{\theta}^*, \mathcal{I}\left(\boldsymbol{\theta}^*\right)^{-1}\right).$$

Normal

假设 $x_1, x_2, ..., x_n$ 是从正态分布 $\mathcal{N}(\mu, \sigma^2)$ 中观测到的数据 (i.i.d)，我们的目标是使用最大似然估计（MLE）来估计未知参数 (μ, σ):

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2}),$$

$$L(\mu,\sigma|x_1,x_2,...,x_n) = (2\pi)^{-\frac{n}{2}}\sigma^{-n}\exp(-\frac{\sum_{i=1}^n(x_i-\mu)^2}{2\sigma^2}),$$

$$\ell(\mu,\sigma|X) = -\frac{n}{2}log(2\pi) - nlog(\sigma) - \frac{\sum_{i=1}^n(x_i-\mu)^2}{2\sigma^2}.$$

然后对其进行求导:

$$\frac{\partial\ell}{\partial\mu} = \frac{\sum_{i=1}^n(x_i-\mu)}{\sigma^2}, \quad \frac{\partial\ell}{\partial\sigma} = -\frac{n}{\sigma} + \frac{\sum_{i=1}^n(x_i-\mu)^2}{\sigma^3}.$$

设 $\nabla \ell = \left(\frac{\partial \ell}{\partial \sigma}, \frac{\partial \ell}{\partial \mu}\right) = (0, 0)$，求解 MLE 估计 $\hat{\mu}$ 和 $\hat{\sigma}$：

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}, \quad \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

接下来求 MLE 的方差，首先先计算观测 Fisher 信息 $(-\nabla^2 \ell)$:

$$\frac{\partial^2 \ell}{\partial \mu^2} = -\frac{n}{\sigma^2}, \quad \frac{\partial^2 \ell}{\partial \mu \partial \sigma} = -\frac{2\sum_{i=1}^{n}(x_i - \mu)}{\sigma^3}$$
$$\frac{\partial^2 \ell}{\partial \sigma^2} = \frac{n}{\sigma^2} - \frac{3\sum_{i=1}^{n}(x_i - \mu)^2}{\sigma^4}$$

求期望之后，得到期望 Fisher 信息

$$\mathcal{I}(\mu, \sigma) = E[-\nabla^2 \ell] = \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{2n}{\sigma^2} \end{bmatrix}, \quad \mathcal{I}^{-1} = \begin{bmatrix} \frac{\sigma^2}{n} & 0 \\ 0 & \frac{\sigma^2}{2n} \end{bmatrix}$$

则 $(\hat{\mu}, \hat{\sigma})$ 的极限分布为 $\mathcal{N}((\mu, \sigma), \mathcal{I}^{-1})$.

值得注意的是，如果我们把 σ^2 当作参数（而不是 σ），则 $\mathcal{I}(\mu, \sigma^2)$ 的计算是不同但相似的：

$$\frac{\partial \ell}{\partial \mu} = \frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^2}, \quad \frac{\partial \ell}{\partial (\sigma)^2} = -\frac{n}{2\sigma^2} + \frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^4},$$
$$\frac{\partial^2 \ell}{\partial \mu^2} = -\frac{n}{\sigma^2}, \quad \frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} = -\frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^4},$$
$$\frac{\partial^2 \ell}{(\partial \sigma^2)^2} = \frac{n}{2\sigma^2} - \frac{\sum_{i=1}^n (x_i - \mu)^2}{\sigma^6}.$$

我们可以通过以下转换来检查结果：

$$\frac{\partial \ell}{\partial \sigma} = \frac{\partial \ell}{\partial (\sigma^2)} \frac{d(\sigma^2)}{d\sigma} = \frac{\partial \ell}{\partial (\sigma^2)} 2\sigma,$$
$$\frac{\partial^2 \ell}{(\partial \sigma)^2} = \frac{\partial (\frac{\partial \ell}{\partial \sigma})}{\partial \sigma} = \frac{\partial (\frac{\partial \ell}{\partial (\sigma)^2})}{\partial (\sigma)^2} 4\sigma^2 + \frac{2\partial \ell}{\partial (\sigma)^2} = \frac{\partial^2 \ell}{(\partial (\sigma)^2)^2} + \frac{2\partial \ell}{\partial (\sigma)^2}.$$

Normal Mixture

有些似然函数太复杂，无法通过分析求解。

假设 $X_1, X_2, \cdots, X_n$ 服从分布：

$$f(x \mid \Theta) = cp\left(x \mid \mu_1, \sigma_1^2\right) + (1-c)p\left(x \mid \mu_2, \sigma_2^2\right),$$

假设 $\Theta = \left(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, c\right)$，且 $p\left(x \mid \mu, \sigma^2\right)$ 表示均值为 μ，方差为 σ^2 的正态分布的密度函数。那么似然函数为：

$$\begin{aligned} L\left(\mu_1, \mu_2, \sigma_1, \sigma_2, c\right) &= \prod_{i=1}^{n} p\left(x_i \mid \mu_1, \mu_2, \sigma_1, \sigma_2, c\right) \\ &= \prod_{i=1}^{n} \left\{ c\frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{\left(x_i-\mu_1\right)^2}{2\sigma_1^2}\right) + (1-c)\frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{\left(x_i-\mu_2\right)^2}{2\sigma_2^2}\right) \right\} \end{aligned}$$

寻找最大似然估计（MLE）→ 优化问题 → 解非线性方程

Summary

Bayesian

贝叶斯学派假设参数是随机。

- 先验分布 $\pi(\Theta)$：关于参数的一些先验信息（例如先前的知识、专家建议）。
- 似然函数 $L(\boldsymbol{\Theta} \mid \boldsymbol{X})$：与频率统计中的定义相同。
- 后验分布 $f(\boldsymbol{\Theta} \mid \boldsymbol{X})$：在观察数据后更新对参数的信息。

根据贝叶斯理论，我们有

$$f(\boldsymbol{\Theta} \mid \boldsymbol{X}) = \frac{f(\boldsymbol{\Theta}, \boldsymbol{X})}{f(x)} = \frac{\pi(\Theta) f(\boldsymbol{X} \mid \boldsymbol{\Theta})}{\int f(\tau, \boldsymbol{X}) d\tau} \propto \pi(\Theta) f(\boldsymbol{X} \mid \boldsymbol{\Theta}).$$

与 Θ 无关的常数，我们称为归一化常数。

分布在差一个常数的情况下，也能够被唯一确定。举例，若 x 满足

$$f(x) \propto exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

设与 x 无关的常数为 c，即 $f(x) = c \cdot exp(-\frac{(x-\mu)^2}{2\sigma^2})$. 则因为定义，我们有

$$\begin{aligned} 1 &= \int c \cdot exp(-\frac{(x-\mu)^2}{2\sigma^2})dx \\ &= c\int exp(-\frac{(x-\mu)^2}{2\sigma^2})dx \\ &= c \cdot \sqrt{2\pi}\sigma \end{aligned}$$

所以我们有 $c = \frac{1}{\sqrt{2\pi}\sigma}$，即正态分布的密度函数。

Conjugacy

在贝叶斯统计中，如果后验分布 $f(\boldsymbol{\Theta} \mid \boldsymbol{X})$ 与先验分布 $\pi(\Theta)$ 属于同类，则先验分布与后验分布被称为共轭分布，而先验分布被称为似然函数的共轭先验。

假设参数 θ 的先验分布为 $\theta \sim N\left(a, b^2\right)$，而数据 $X_1, \cdots, X_n$ (i.i.d) 的分布为 $N(\theta, 1)$。

$$
\begin{aligned}
f(\theta \mid \mathbf{x}) \propto f(\mathbf{x} \mid \theta)\pi(\theta) &= \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_i-\theta)^2}{2}\right)\right] \frac{1}{\sqrt{2\pi}b} \exp\left(-\frac{(\theta-a)^2}{2b^2}\right) \\
&\propto \exp\left(-\frac{n\theta^2 - 2\theta\sum_{i=1}^{n} x_i}{2} - \frac{\theta^2 - 2a\theta}{2b^2}\right) \\
&\propto \exp\left(-\frac{\left(nb^2+1\right)\theta^2 - 2\left(nb^2\bar{x}+a\right)\theta}{2b^2}\right) \\
&\propto \exp\left(-\frac{\left(\theta - \left(nb^2\bar{x}+a\right)/\left(nb^2+1\right)\right)^2}{2b^2/\left(nb^2+1\right)}\right)
\end{aligned}
$$

$$f(\theta \mid \mathbf{x}) \propto \exp\left(-\frac{\left(\theta-\left(nb^2\bar{x}+a\right)/\left(nb^2+1\right)\right)^2}{2b^2/\left(nb^2+1\right)}\right)$$

因此，后验分布为 $\theta \mid \boldsymbol{X} \sim N\left(\eta, \tau^2\right)$，其中

$$\eta=\frac{\bar{x}+a/(nb^2)}{1+1/(nb^2)}, \quad \tau^2=\frac{1}{n+1/b^2}.$$

n 越大，样本数量越多，先验分布的影响就越小。

当先验信息较为不准确时，重要的是确保所选择的先验分布不会对后验推断产生很大的影响。（可以选择一个较大的先验方差，也就是 b 值）

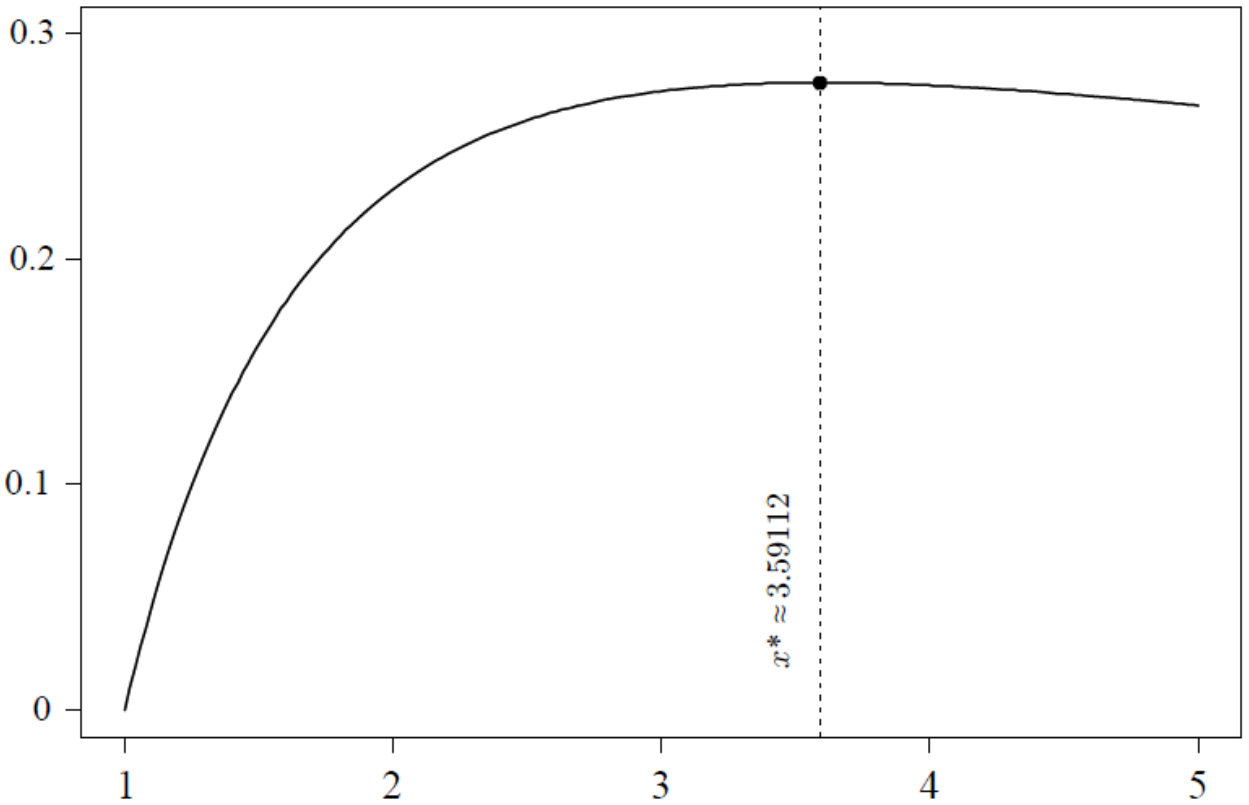
贝叶斯学派下求参数，通常是用 MCMC 采样出后验分布下的参数样本。当然我们也可以用最大化后验分布的形式，求得参数的点估计，即后验分布的 mode.

Summary

Univariate Problem

若要最大化 $g(x)$

$$g(x) = \frac{logx}{1+x}, \quad g'(x) = \frac{1+1/x-logx}{(1+x)^2}.$$



Bisection Method (Find $g'(x) = 0$)

引理：如果 g' 在 $[a_0, b_0]$ 上连续且满足 $g'(a_0)g'(b_0) \leq 0$，那么中值定理表明在 $[a_0, b_0]$ 上至少存在一个 $x^* \in [a_0, b_0]$，使得 $g'(x) = 0$，因此 x^* 是 g 的一个局部极值点。

为了找到这个点，二分法系统地将区间从 $[a_0, b_0]$ 缩小到 $[a_1, b_1]$，再缩小到 $[a_2, b_2]$，依此类推，其中 $[a_0, b_0] \supset [a_1, b_1] \supset [a_2, b_2] \supset \cdots$ 等等。

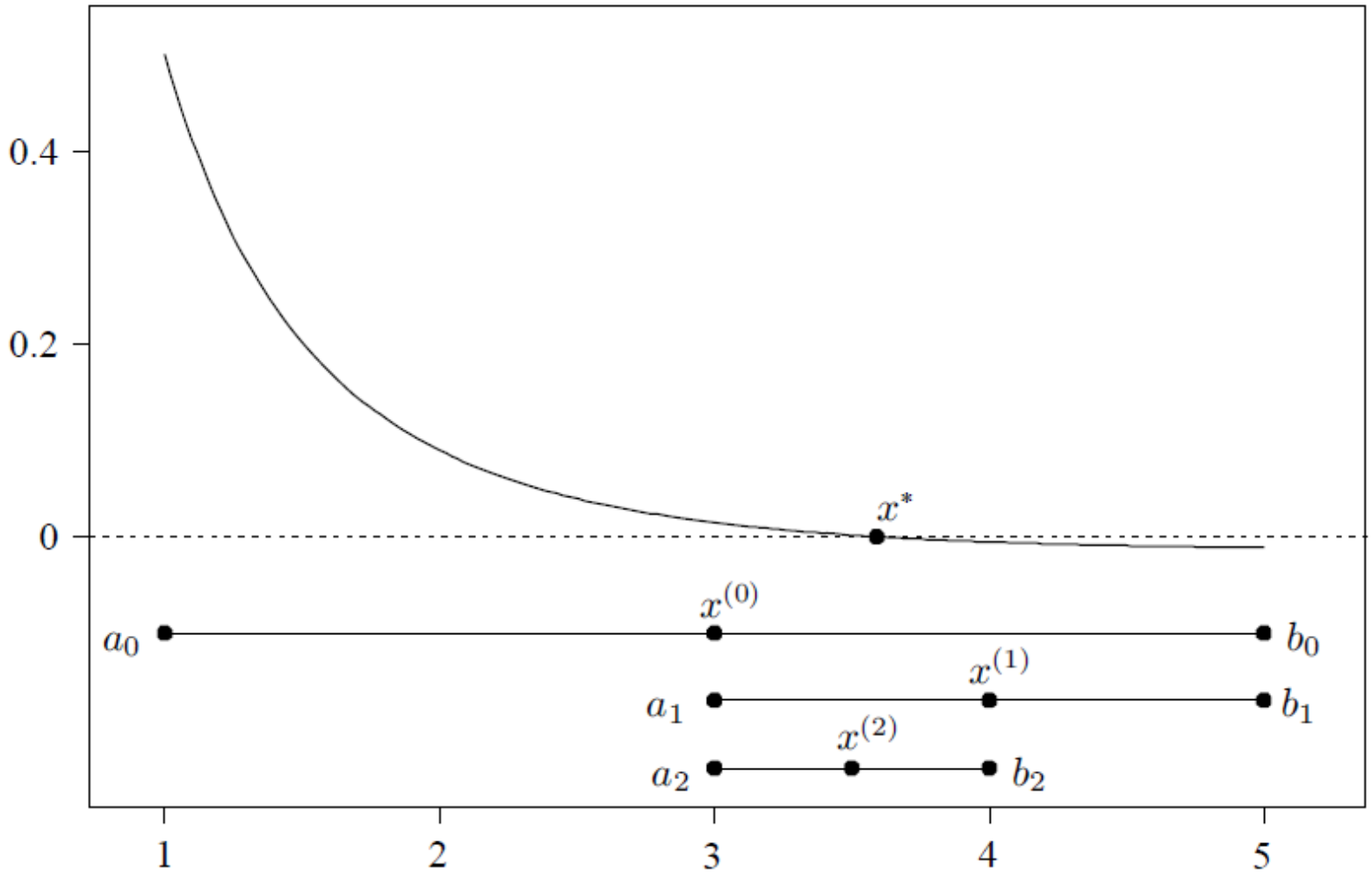
令 $x^{(0)} = (a_0 + b_0)/2$ 为初始值. 则更新公式为

$$[a_{t+1}, b_{t+1}] = \begin{cases} [a_t, x^{(t)}] & \text{if } g'(a_t)\, g'(x^{(t)}) \leq 0, \\ [x^{(t)}, b_t] & \text{if } g'(a_t)g'(x^{(t)}) > 0 \end{cases}$$

和

$$x^{(t+1)} = \frac{1}{2}(a_{t+1} + b_{t+1}).$$

为了找到使 $g(x)$ 最大化的 x 值，即 $g'(x) = 0$ 的值，我们可以取 $a_0 = 1$，$b_0 = 5$，和 $x^{(0)} = 3$。



Summary of Bisection Method

二分法是一种特殊的括号法（bracketing method），也就是说，它通过一系列逐渐缩小长度的嵌套区间来限定一个根的范围。

- 二分法是一种相对较慢的方法：它需要较多的迭代次数才能达到所需的精度。
- 如果 g' 在 $[a_0, b_0]$ 上连续，无论 g'' 是否存在、求导的难易性如何，都可以找到一个根。

Summary

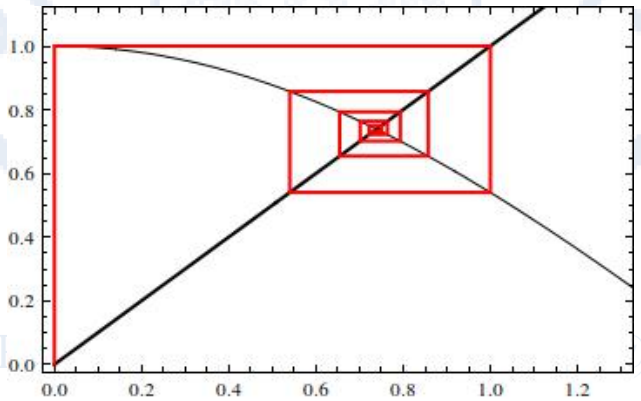
Fixed-Point Iteration (Functional Iteration)

寻找根的不动点策略是确定一个函数 G，满足当且仅当 $g'(x) = 0$ 时 $G(x) = x$。最简单方法是使用更新方程 $x^{(t+1)} = G(x^{(t)})$。

G 的选择不是固定的，但最明显的选择是 $G(x) = g'(x) + x$:

$$x^{(t+1)} = x^{(t)} + g'(x^{(t)})$$

例子: 求解 $g'(x) = cosx - x$ 的根, 我们设置 $G(x) = cosx$, $x_0 = 0$:



Convergence condition

该算法的收敛性取决于函数 G 在 $[a,b]$ 上是否是收缩的:

1. $\forall x \in [a,b]$, **我们有** $G(x) \in [a,b]$,
2. $\exists \lambda \in [0,1)$, **使得**

$$|G(x_1) - G(x_2)| \leq \lambda |x_1 - x_2|$$

对任意 $x_1, x_2 \in [a,b]$. (*Lipschitz Condition*)

则我们有

a. $G(x)$ **有一个唯一的固定点** $x_\infty \in [a,b]$,

b. $x_n = G(x_{n-1}) \to x_\infty, \quad \forall x_0 \in [a,b]$,

c. $|x_n - x_\infty| \leq \frac{\lambda^n}{1-\lambda}|x_1 - x_0|$.

Proof: 根据条件（1）和（2），我们有

$$|x_{k+1} - x_k| = |G(x_k) - G(x_{k-1})| \leq \lambda |x_k - x_{k-1}| \leq \cdots \leq \lambda^k |x_1 - x_0|,$$

因此 $\forall m, n$, 我们有

$$|x_m - x_n| \leq \sum_{k=n}^{m-1} |x_{k+1} - x_k| \leq \sum_{k=n}^{m-1} \lambda^k |x_1 - x_0| \leq \frac{\lambda^n}{1-\lambda} |x_1 - x_0|.$$

因此，该序列是一个柯西序列，而且由于 $[a, b]$ 是闭区间，我们有（b）$x_n \to x_\infty \in [a, b]$。令 $m \to \infty$，我们有（c）$|x_n - x_\infty| \leq \frac{\lambda^n}{1-\lambda} |x_1 - x_0|$。

要注意的是，上述证明中，条件（1）其实也起了作用。

又由于 G 是连续的，

$$x_\infty = \lim_{n\to\infty} x_n = \lim_{n\to\infty} G(x_{n-1}) = G(\lim_{n\to\infty} x_{n-1}) = G(x_\infty),$$

因此 x_∞ 是 $G(x)$ 的一个不动点。

对于唯一性，假设存在 $y_\infty \neq x_\infty$，使得 $G(y_\infty) = y_\infty$，那么

$$|y_\infty - x_\infty| \quad = |G(y_\infty) - G(x_\infty)| \leq \lambda |y_\infty - x_\infty| < |y_\infty - x_\infty|,$$

矛盾，所以只存在唯一。

Sufficient condition for Lipschitz Continuity

如果对于 $[a, b]$ 中的所有 x，都满足 $|G'(x)| \leq \lambda < 1$，则满足了 Lipschitz 条件。

Proof: 对于任意的 x, y，根据均值定理（G 可微分），存在 $\xi \in [y, x]$，使得 $G(x) - G(y) = G'(\xi)(x - y)$，因此

$$|G(x) - G(y)| = |G'(\xi)||x - y| \leq \lambda |x - y| < |x - y|.$$

Lipschitz 连续函数限制了函数变化的速度。具体而言，满足 Lipschitz 条件的函数的斜率必须受到一个实数的限制，该实数被称为 Lipschitz 常数。

Scaling for non-convergence

如果 $G(x) = g'(x) + x$，则推论要求在 $[a, b]$ 上满足 $|g''(x) + 1| < 1$。当 g'' 有界且在 $[a, b]$ 上不改变符号时，我们可以通过选择 $G(x) = \alpha g'(x) + x$ 和 $|\alpha g''(x) + 1| < 1$ 来重新缩放非收敛问题。更新公式为：

$$x_{n+1} = x_n + \alpha g'(x_n).$$

不动点迭代的效果高度依赖于所选择的 G 的形式。例如，考虑找到方程 $g'(x) = \log x + x$ 的根。不同 G 的收敛速度如下：

$$(x + e^{-x})/2 > e^{-x} > -\log x$$

接下来，我们将介绍函数迭代的特殊情况：牛顿法和割线法。

Summary

Newton's Method (Newton–Raphson method)

假设 g' 是连续可微的，并且 $g''(x) \neq 0$。在第 t 次迭代中，该方法通过线性泰勒级数展开来近似 $g'(x)$：

$$0 = g'(x) \approx g'(x^{(t)}) + (x - x^{(t)})g''(x^{(t)}).$$

上式右端即为 g' 在 $x^{(t)}$ 处的切线，我们用切线的根来近似 g' 的根。因此，解上述方程得到 x^* 的近似值：

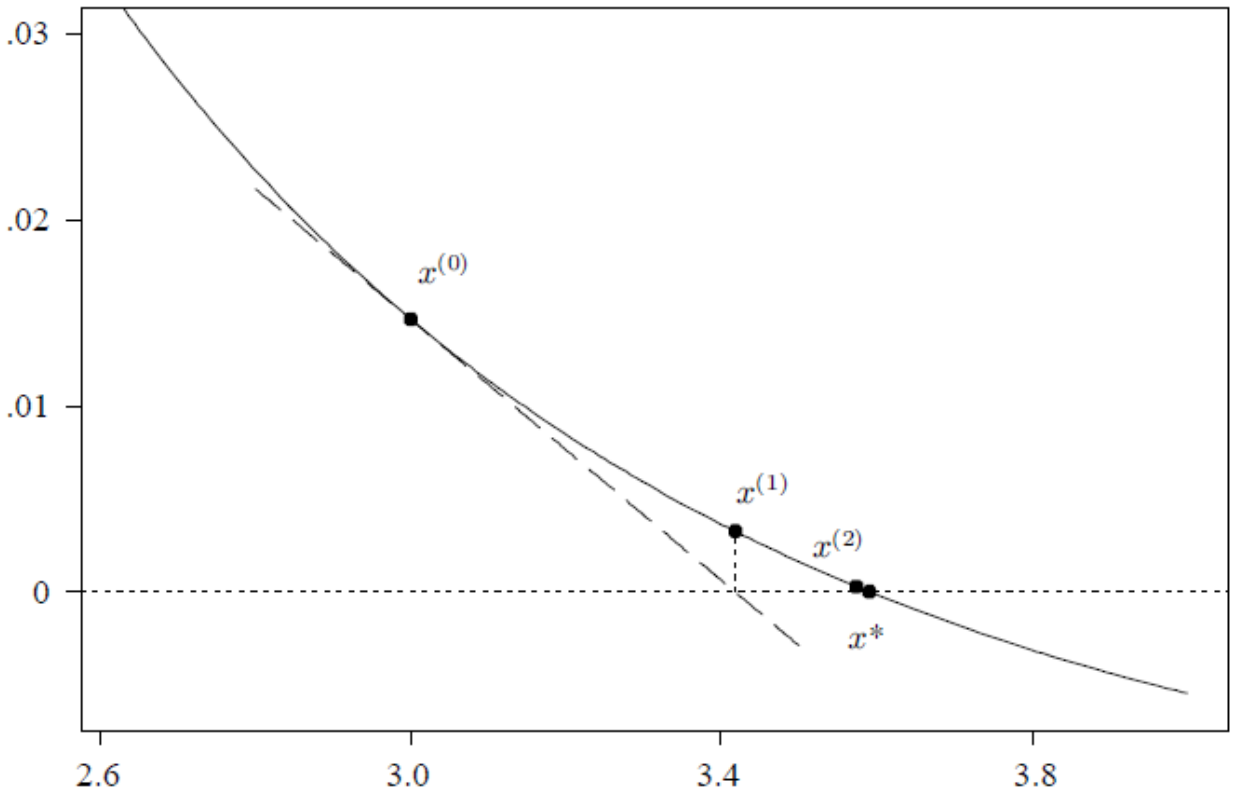
$$x^* = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})} = x^{(t)} + h^{(t)}.$$

这个方程描述了一个依赖于当前猜测 $x^{(t)}$ 和修正项 $h^{(t)}$ 的 x^* 的近似值。迭代这个策略给出了牛顿法的更新方程：

$$x^{(t+1)} = x^{(t)} + h^{(t)}$$

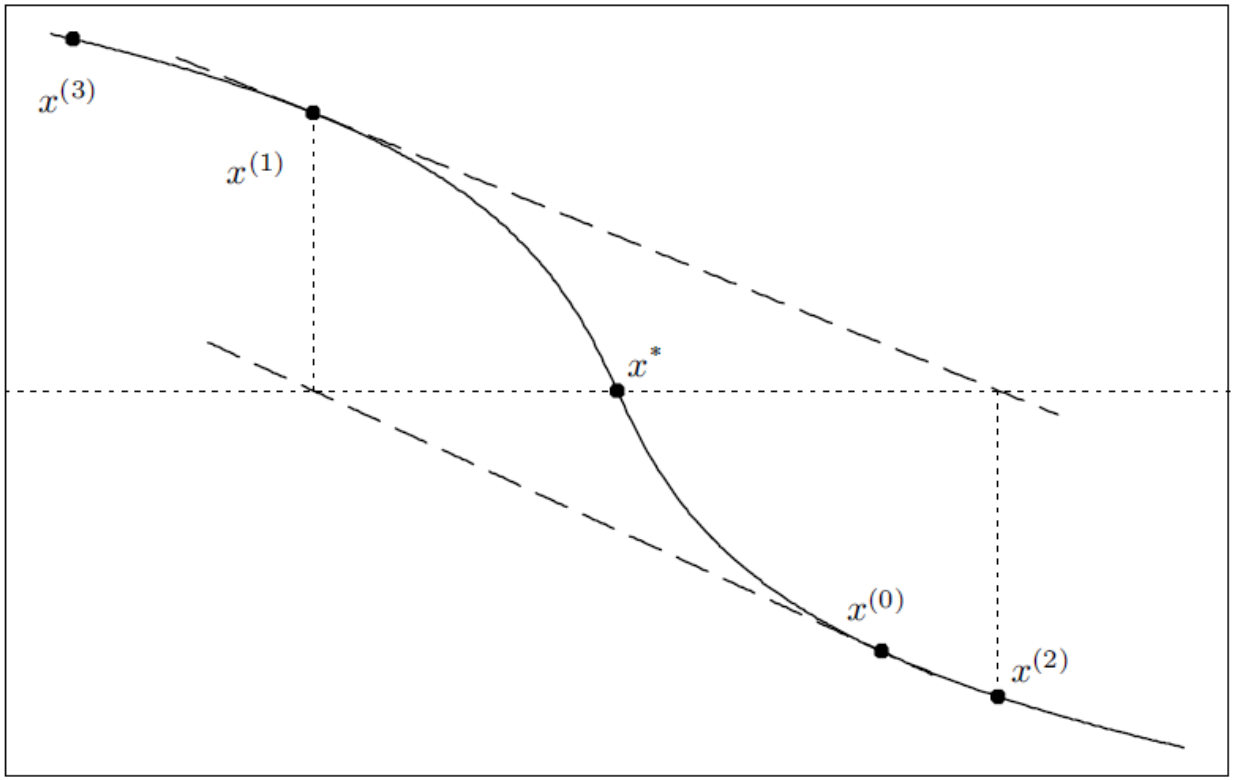
$$y = g'(x^{(t)}) + (x - x^{(t)})g''(x^{(t)})$$

$$y = 0 \rightarrow x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})}$$



Newton's Method Diverges

牛顿法的收敛性取决于函数 g 的形状和初始值的选择。



Convergence in the neighborhood of x^*

（根附近的收敛性）定理: 如果 g''' 连续且 $g'(x^*) = 0$，$g''(x^*) \neq 0$，那么存在一个 x^* 的邻域，在该邻域内从任意 $x_0 \in (x^* - \delta, x^* + \delta)$ 开始的牛顿法迭代

$$x_n = G(x_{n-1}) = x_{n-1} - \frac{g'(x_{n-1})}{g''(x_{n-1})}$$

会收敛到 x^*。

证明: 求导可得 $G'(x^*) = 0$ (注意 $G(x)$ 不是切线！) 并且 $G'(x)$ 是连续的。因此，给定 $\lambda < 1$，存在 $\delta > 0$，对于所有 $x \in (x^* - \delta, x^* + \delta)$，我们有 $|G'(x)| < \lambda$。因此，该函数满足 Lipschitz 连续性（常数 < 1）。

由不动点存在，易证对所有 $x \in (x^* - \delta, x^* + \delta)$，我们有 $G(x) \in [x^* - \delta, x^* + \delta]$. 所以函数有收缩性，收敛性条件得以证毕。

Global Convergence

当 g' 具有二阶连续可导性、是凸函数并且根存在的时候，牛顿法可以从任意起始点收敛到该根。

凸函数: 如果一个实值函数的图像上任意两点间的线段位于这两点之间的图像上方，则该函数被称为凸函数。对任意 $0 \leq t \leq 1$ 并且任意 $x_1, x_2 \in X$:

$$f\left(tx_1 + (1-t)x_2\right) \leq tf\left(x_1\right) + (1-t)f\left(x_2\right)$$

一个单变量的二次可微函数是凸函数，当且仅当它的二阶导数在整个定义域上非负。

Convergence in an interval

当从区间 $[a, b]$ 中的某个点开始时，可以检查的另一组条件如下所示。

1. $g''(x) \neq 0$ 在 $[a, b]$ 上,
2. $g'''(x)$ 在 $[a, b]$ 不变符号,
3. $g'(a)g'(b) < 0$, 并且
4. $|g'(a)/g''(a)| < b - a$ 并且 $|g'(b)/g''(b)| < b - a$,

那么牛顿法将从区间内的任意 x_0 收敛。

Convergence Order

牛顿法等求根方法的速度通常通过其收敛阶数来衡量。如果一个方法具有收敛阶数 β，则当 n 趋向无穷时，满足 $\lim_{n\to\infty}|x_n - x^*| = 0$ 以及

$$\lim_{n\to\infty}\frac{|x_n - x^*|}{|x_{n-1} - x^*|^\beta} = c$$

其中 $c \neq 0$ 和 $\beta > 0$ 是一些常数。对于牛顿法，它具有二次收敛，即 $\beta = 2$:

$$\begin{aligned} x_n - x^* = G(x_{n-1}) - G(x^*) &= G'(x^*)(x_{n-1} - x^*) + \frac{1}{2}G''(z_n)(x_{n-1} - x^*)^2 \\ &= \frac{1}{2}G''(z_n)(x_{n-1} - x^*)^2, \end{aligned}$$

$$\lim_{n\to\infty}\frac{|x_n - x^*|}{|x_{n-1} - x^*|^2} = \lim_{n\to\infty}\frac{G''(z_n)}{2} = \frac{G''(x^*)}{2}$$

而二分法是线性收敛的 $(\beta = 1)$.

Fisher Scoring

回想一下，$\mathcal{I}(\theta)$ 可以近似为 $-\ell''(\theta)$。因此，Fisher Scoring 方法：

$$\theta^{(t+1)} = \theta^{(t)} + \ell'(\theta^{(t)})\mathcal{I}(\theta^{(t)})^{-1}$$

Fisher Scoring 方法在开始阶段可以更好地实现快速改进，而牛顿法在接近结束时的细化过程中表现更好。

Multivariate Probelm

在变量多维的时候，对于牛顿法，我们再次使用一阶泰勒级数展开来近似 $g'(\boldsymbol{X}^*)$：

$$g'(\boldsymbol{X}^*) \approx g'(\boldsymbol{X}^{(t)}) + (\boldsymbol{X}^* - \boldsymbol{X}^{(t)})^{\mathrm{T}} g''(\boldsymbol{X}^{(t)})$$

则我们有更新公式：

$$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} - g''(\boldsymbol{X}^{(t)})^{-1} g'(\boldsymbol{X}^{(t)})$$

在最大似然估计中，我们可以用 $\mathcal{I}(\Theta(t))$，即在 $\Theta(t)$ 处的期望 Fisher 信息，来替代 $\Theta(t)$ 处的观测信息：

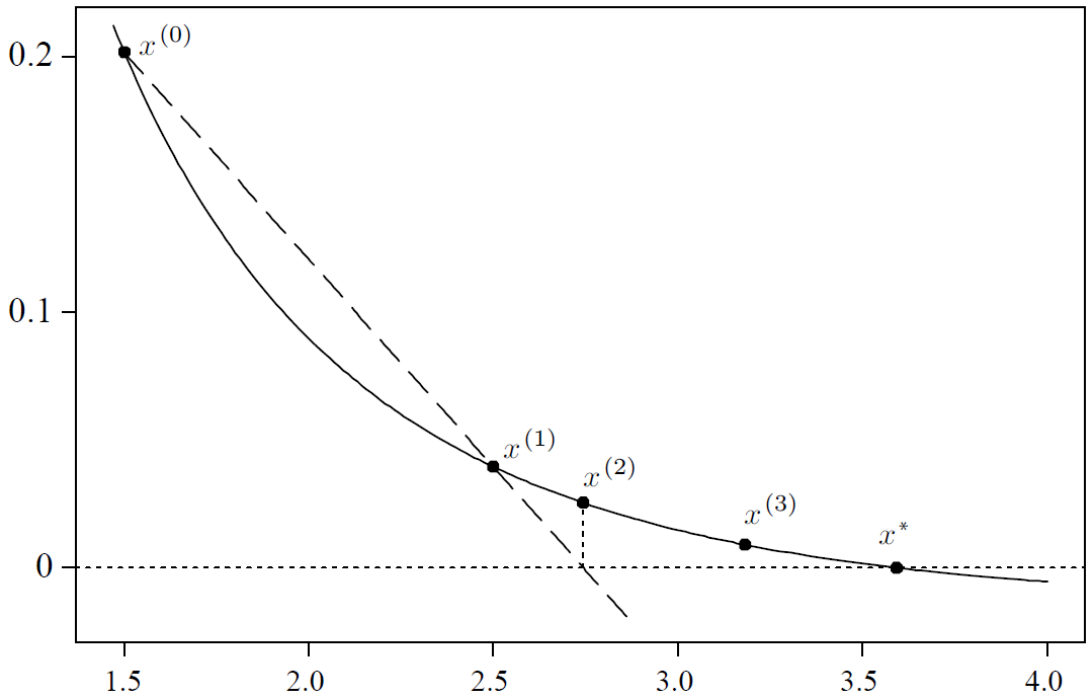
$$\Theta^{(t+1)} = \Theta^{(t)} + \mathbf{I}(\theta^{(t)})^{-1} \ell'(\Theta^t).$$

Summary

Secant Method

除牛顿法以外，单变量的求根问题也可以用割线法。我们使用 $[g'(x^{(t)}) - g'(x^{(t-1)})]/(x^{(t)} - x^{(t-1)})$ 来替换 $g''(x)$。

$$x^{(t+1)} = x^{(t)} - g'(x^{(t)}) \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}$$



Newton-Like Methods

一些非常有效的方法都依赖于以下形式的更新方程：

$$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} - (\mathbf{M}^{(t)})^{-1} \nabla g(\boldsymbol{X}^{(t)})$$

其中 $\mathbf{M}^{(t)}$ 是一个近似海森矩阵 $\nabla^2 g(\boldsymbol{X}^{(t)})$ 的 $p \times p$ 矩阵。我们已经知道一种可能的海森矩阵替代方法，即 $\mathbf{M}^{(t)} = -\mathcal{I}(\Theta(t))$，可以得到 Fisher Scoring 方法。

最速上升算法： $\mathbf{M}^{(t)} = -I$：

$$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} + \nabla g(\boldsymbol{X}^{(t)})$$

沿着最陡上升方向进行缩放步长：

$$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} + \alpha^{(t)} \nabla g(\boldsymbol{X}^{(t)})$$

Ascent

我们做上述优化算法的时候，可迫使它的每一步，也就是每一次更新，目标函数都是上升的。

对于任何固定的 $x^{(t)}$ 和负定 $\mathbf{M}^{(t)}$，请注意 $\alpha^{(t)} \to 0$ 我们有

$$\begin{aligned} g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^{(t)}) &= g(\mathbf{x}^{(t)} + \mathbf{h}^{(t)}) - g(\mathbf{x}^{(t)}) \\ &= -\alpha^{(t)} g'(\mathbf{x}^{(t)})^{\mathrm{T}} (\mathbf{M}^{(t)})^{-1} g'(\mathbf{x}^{(t)}) + o(\alpha^{(t)}) \end{aligned}$$

其中 $h^{(t)} = -\alpha^{(t)}[\mathbf{M}^{(t)}]^{-1}\nabla g(\boldsymbol{X}^{(t)})$，第二个等式来自于 Taylor 展开 $g(x^{(t)} + h^{(t)})) = g(x^{(t)}) + g'(x^{(t)})^T h^{(t)} + o(\alpha^{(t)})$

因此，如果 $-\mathbf{M}(t)$ 是正定的，通过选择足够小的 $\alpha(t)$，可以确保上升，从而得到 $g(x(t+1)) - g(x(t)) > 0$。

牛顿法并不能确保每次迭代目标函数都是上升的，而且计算成本大。合理的选择更新形式，比如最速上升算法，有可能会得到更好的表现，以及更小的计算量。

Summary

Optimization in Deep Learning

我们试图最小化平均损失函数：

$$g(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$$

其中 $\ell_i(x)$ 是损失函数，而不是对数似然函数。

1. GD (Gradient Descent): $x_t = x_{t-1} - \alpha \nabla g(x_{t-1})$
2. SGD (Stochastic Gradient Descent): $x_t = x_{t-1} - \alpha \nabla \ell_{t_i}(x_{t-1})$
3. **BGD (Batch Gradient Descent)**: $x_t = x_{t-1} - \frac{\alpha}{|\mathcal{B}|}\sum_{i\in\mathcal{B}_t} \nabla \ell_i(\boldsymbol{x}_{t-1})$

我们常常使用批量梯度下降法（batch gradient descent）。

Epoch vs Iteration

在神经网络术语中：

1. one epoch = 进行一次前向传播和一次反向传播，针对所有的训练样本。
2. batch size = 在一次前向传播/反向传播中的训练样本数量。批量大小越大，所需的内存空间就越多。
3. number of iterations = 进行的遍数，每次遍数使用 [批量大小] 个样本。明确地说，一次遍数 = 一次前向传播 + 一次反向传播（我们不将前向传播和反向传播视为两个不同的遍数）。

例如：如果你有 1000 个训练样本，batch size 为 500，那么完成一个 epoch 需要 2 次 iterations。

Momentum

SGDM（Stochastic Gradient Descent with Momentum）是对随机梯度下降（SGD）的扩展，它引入动量来加速收敛并减小震荡。

$$g_t = \sum_{i \in \mathcal{B}_t} \nabla \ell_i(\boldsymbol{x}_{t-1})$$

$$v_t = \beta v_{t-1} + g_t, \quad x_t = x_{t-1} - \alpha v_t$$

梯度的指数加权移动平均：

$$v_t = g_t + \beta g_{t-1} + \beta^2 g_{t-2} + \beta^3 g_{t-3} + \ldots$$

β 常用的值为：[0.5, 0.9, 0.95, 0.99].

AdaGrad

AdaGrad 是一种自适应学习率优化算法，它根据历史梯度调整每个参数的学习率。类似标准化。

$$s_t = s_{t-1} + g_t \cdot g_t, \quad x_t = x_{t-1} - \frac{\alpha}{\sqrt{s_t} + \epsilon} \cdot \nabla g_t.$$

其中 $\cdot$ 表示逐元素相乘。

RMSProp ((Root Mean Square Propagation))

RMSProp 是一种自适应学习率优化算法，通过引入衰减因子来解决 AdaGrad 的一些局限性。

$$s_t = \gamma s_{t-1} + (1-\gamma) g_t \cdot g_t, \quad x_t = x_{t-1} - \frac{\alpha}{\sqrt{s_t} + \epsilon} \cdot g_t.$$

Adam (Adaptive Moment Estimation)

Adam 是一种自适应学习率优化算法，结合了 Momentum 和 RMSProp 的优点。

$$v_t = \beta_1 v_{t-1} + (1-\beta_1) g_t, \quad s_t = \beta_2 s_{t-1} + (1-\beta_2) g_t \cdot g_t$$

常见的值是 $\beta_1 = 0.9$，$\beta_2 = 0.999$。由于当 t 受限时，指数加权和不等于 1，因此我们进行偏差修正：

$$\hat{v}_t = \frac{v_t}{1-\beta_1^t}, \quad \hat{s}_t = \frac{s_t}{1-\beta_2^t}$$

最终更新公式是

$$x_t = x_{t-1} - \alpha \frac{\hat{v}_t}{\sqrt{\hat{s}_t} + \epsilon}$$

请注意，Adam 不是 AdaGrad。

References

- 7.8 Adam Algorithm (Dive into DL PyTorch)
- Gentle Introduction to the Adam Optimization Algorithm for Deep Learning (Blog)
- Adam Method (Papers with Code)