高斯分布的最大似然估计

emacsun

目录

1 原理

给定一个数据集$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$。假设$\{\mathbf{x}_n\}$是多变量高斯分布的一个独立的观察。我们可以通过最大似然估计来估计高斯分布的参数。对数似然函数是：

$$\ln p(\mathbf{X}|\mu, \mathbf{\Sigma}) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_n - \mu) \quad (1.1)$$

对上式化简，我们发现似然函数对数据集合的依赖体现在$\sum_{n=1}^{N}\mathbf{x}_n$和$\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T$两个量上。这两个量叫做高斯分布的充分统计量（sufficient statistics）。不同的分布有不同的充分统计量，这个我们用到的时候在详谈，此处不展开。

在式 (1.1)中，对μ求导，有：

$$\frac{\partial}{\partial\mu}\ln p(\mathbf{X}|\mu, \mathbf{\Sigma}) = \sum_{n=1}^{N}\sum_{n=1}^{N}\mathbf{\Sigma}^{-1}(\mathbf{x}_n - \mu) \quad (1.2)$$

令上式为零，则我们得到了关于高斯分布均值的最大似然解：

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \quad (1.3)$$

显然，这个最大似然解是观测数据集合的均值。

对 (1.1)的$\mathbf{\Sigma}$求导，有：

$$\mathbf{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \mu_{ML})(\mathbf{x}_n - \mu_{ML})^T \quad (1.4)$$

式~(1.4)中出现了μ_{ML}，这是联合优化μ和$\mathbf{\Sigma}$的结果。另外注意到μ_{ML}与$\mathbf{\Sigma}_{ML}$无关，所以我们可以先得到μ_{ML}，然后求$\mathbf{\Sigma}_{ML}$。

基于μ_{ML}和$\mathbf{\Sigma}_{ML}$，我们求高斯分布的期望和方差：

$$\mathbb{E}[\mu_{ML}] = \mu \tag{1.5}$$

$$\mathbb{E}[\mathbf{\Sigma}_{ML}] = \frac{N-1}{N}\mathbf{\Sigma} \tag{1.6}$$

我们发现最大似然估计的均值等于真实的均值，最大似然估计的方差总是小于真实值，因此这个估计是有偏的（biased）.我们可以定义一个不同的估计：

$$\tilde{\mathbf{\Sigma}} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \mu_{ML})(\mathbf{x}_n - \mu_{ML})^T \tag{1.7}$$

显然$\tilde{\mathbf{\Sigma}}$的期望与$\mathbf{\Sigma}$相等。

2 应用

以上讨论高斯分布参数的最大似然估计，这个过程为我们进行序贯估计（sequential estimation）提供了方便。序贯算法允许数据在线处理。所谓在线处理（on-line process）是指一次处理一个数据点然后丢点这个数据点。在线处理的优势是相对于离线处理（off-line）在线处理可以不用一次性保存并处理大量的数据。

考虑式 (1.3)，对高斯分布均值的最大似然估计，如果我们把式(1.3)写成递推的形式，则有：

$$\mu_{ML}^{(N)} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \tag{2.1}$$

$$= \frac{1}{N}\mathbf{x}_N + \frac{1}{N}\sum_{n=1}^{N-1}\mathbf{x}_n \tag{2.2}$$

$$= \frac{1}{N}\mathbf{x}_N + \frac{N-1}{N}\mu_{ML}^{(N-1)} \tag{2.3}$$

$$= \mu_{ML}^{(N-1)} + \frac{1}{N}(\mathbf{x}_N - \mu_{ML}^{(N-1)}) \tag{2.4}$$

这个结果提供了一个递推的求解高斯分布均值的方法。接收到第$N-1$个数据之后，我们对μ的估计$\mu_{ML}^{(N-1)}$。我们现在观察到了$\mathbf{x}_N$，那么我们基于$\mathbf{x}_N$和$\mu_{ML}^{(N-1)}$得到一个更新的$\mu_{ML}^{(N-1)}$。仔细观察这个结果，我们发现相对于$\mu_{ML}^{(N-1)}$，更新的$\mu_{ML}^{(N)}$在原来的基础上更新了一个很小的量$\frac{1}{N}(\mathbf{x}_N - \mu_{ML}^{(N-1)})$。

式 (2.1)和式(1.3)在本质上是相同的，提供了一种迭代计算均值的方法。但是在实际中我们却较少使用这种方法，我们更general的序贯学习方法。Robbins-Monro算法就是比较general的算法。考虑一对随机变量θ和z，其联合概率分布是$p(z,\theta)$.那么，给定θ求z的条件期望确定了$f(\theta)$：

$$f(\theta) = \mathbb{E}[z|\theta] = \int zp(z|\theta)\mathrm{d}z \tag{2.5}$$

式 (2.5) 的结果可以用图1来表示。

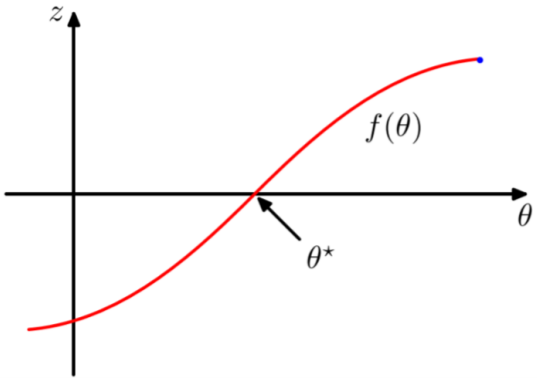

图 1: Robbins-Monro算法

通过式~(2.5)定义的函数叫做回归函数(regression functions). 定义了式 (2.5)之后，我们的目标是找到θ^*使得$f(\theta^*) = 0$。对于z和θ，如果我们有一个较大的数据集。我们可以直接获得回归函数，并估计它的零点。

假设我们观测到了z的一个样本，然后我们期望得到对应的θ^*的序贯估计。Robbins-Monro提供了一个过程。假设:

$$\mathbb{E}[(z-f)^2|\theta] < \infty \tag{2.6}$$

另外，不失一般性，我们认为$f(\theta) > 0, \theta > \theta^*$, 且$f(\theta) < 0, \theta < \theta^*$, 就像图1所示的那样。Robbins=Monro过程定义了估计θ^*的一个递推式：

$$\theta^{(N)} = \theta^{(N-1)} + a_{N-1}z(\theta^{(N-1)}) \tag{2.7}$$

其中$z(\theta^{(N)})$是当θ取值$\theta^{(N)}$时z的一个观测值。系数$\{a_N\}$代表一系列正数，满足：

$$\lim_{N\to\infty} a_N = 0 \tag{2.8}$$

$$\sum_{N=1}^{\infty} a_N = \infty \tag{2.9}$$

$$\sum_{N=1}^{\infty} a_N^2 < \infty \tag{2.10}$$

Robbins和Monro证明了式 (2.7)给出序贯估计的确可以以概率1收敛到θ^*。

现在让我们仔细考虑使用Robbins-Monro算法如何可以让一个广义的最大似然估计问题收敛。我们知道，一句定义最大似然估计解θ_{ML}是对数似然函数的一个静态点，满足：

$$\frac{\partial}{\partial\theta}\left\{\frac{1}{N}\sum_{n=1}^{N}\ln p(\mathbf{x}_n|\theta)\right\}|_{\theta_{ML}} = 0 \tag{2.11}$$

交换积分和求导顺序，令$N\to\infty$,我们有：

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial\theta}\ln p(x_n|\theta) = \mathbb{E}[\frac{\partial}{\partial\theta}\ln p(x|\theta)] \tag{2.12}$$

因此我们看到找到最大似然解相当于找到回归函数的根。