Table Of Contents：

今日工作

1、阅读《计算机视觉 模型、学习合推理》第四章：拟合概率模型 和第五章：正态分布 。

2、分析之前代码中一些不太明白的函数的功能。

拟合概率模型

*x1,x2, ... ,xI*是从独立抽样得到的数据。

`x*` 是新数据点，讨论 `x*` 在拟合的概率模型下的概率。

`θ*` 是拟合的概率模型的参数。

最大似然法(ML)

当*Pr(x1...I|θ)*最大时，这个*θ*就是 `θ*` 。

这个方法理解为，为*θ*赋值，计算在这个条件下，抽样得到*x1,x2, ... ,xk*这组数据的概率。

当这个概率最大时，表示原分布与*θ*条件下的分布非常相似。

数据点 `θ*` 的估计概率密度是 `Pr(x*|θ*)` 。

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\left[\operatorname{Pr}\left(\boldsymbol{x}_{1 \ldots I} \mid \boldsymbol{\theta}\right)\right] \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\left[\prod_{i=1}^{I} \operatorname{Pr}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\theta}\right)\right]
\end{aligned}
$$

最大后验法(MAP)

最大后验估计是最大化参数的后验概率*Pr(θ|x1...I)*。

在MAP拟合中，引入了参数*θ*的先验信息*Pr(θ)*。

因为*Pr(x1...I)*与*θ*无关，因此最大值位置与分母无关，则

$$
\hat{\boldsymbol{\theta}} = \underset{\theta}{\operatorname{argmax}}\left[\prod_{i=1}^{I} \operatorname{Pr}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\theta}\right) \operatorname{Pr}(\boldsymbol{\theta})\right]
$$

最大似然法是最大后验法在先验信息*Pr(θ)*未知情况下的特例。

在最大后验法中，分布函数的参数*θ*的分布函数的参数是给定的。

例如正态分布，它的共轭正态逆伽马分布的参数α，β，γ，δ是给定的。

贝叶斯方法

贝叶斯方法基于一个事实：

有多个参数*θ*符合抽样数据的分布。

基于上述的多个参数*θ*，估计点 `x*` 的概率密度：

$$\Pr(\boldsymbol{x}^* \mid \boldsymbol{x}_{1...I}) = \int \Pr(\boldsymbol{x}^* \mid \boldsymbol{\theta}) \Pr(\boldsymbol{\theta} \mid \boldsymbol{x}_{1...I}) \mathrm{d}\boldsymbol{\theta}$$

其中，*Pr(θ|x1... xI)*的值为：

$$\Pr(\boldsymbol{\theta} \mid \boldsymbol{x}_{1...I}) = \frac{\prod_{i=1}^{I} \Pr(\boldsymbol{x}_i \mid \boldsymbol{\theta}) \Pr(\boldsymbol{\theta})}{\Pr(\boldsymbol{x}_{1...I})}$$

注意，这个公式中出现了共轭性。先验信息*Pr(θ)*的值由分布函数的共轭给出。

对于正态分布：

其中，*k*是一个常数。

这里对应了前一章关于共轭性的内容。

示例

最大似然法

以正态分布的数据举例：

可以看出，图c中的参数是对抽样数据拟合得最好的。

`L = Pr(x1 ... xI|θ)` 在 `+` 位置取得最大值。同时也是极值。

总结

书中介绍了三种拟合方法，针对每种方法，给出了对应的算例，通过算例，对拟合过程有了更清晰的认识。

总的来说，拟合是根据已有的抽样数据，求新数据出现的概率。

`ML` 合 `MAP` 方法需要求得这个拟合的分布的具体参数 `θ`，

`贝叶斯方法` 基于条件：有多个 `θ` 符合抽样数据，然后用类似加权的方法，求得新数据出现的概率。

代码分析

最主要的工作是手动将样本整理为一批，让其能够输入模型。

使用四张三通道图像，手动构建一个batch。

并将其与通过DataLoader获得的batch对比。

除此之外还进行了*zip()*等函数的分析，内容比较杂，想到什么不理解内容的就测试了一下。