贝叶斯推理

贝叶斯推理[1]（Bayesian inference）是统计学中的一个重要问题，也是许多机器学习方法中经常遇到的问题。例如，用于分类的高斯混合模型或用于主题建模的潜在狄利克雷分配（Latent Dirichlet Allocation，简称LDA）模型等概率图模型都需要在拟合数据时解决这一问题。

同时，由于模型设置（假设、维度……）不同，贝叶斯推理问题有时会很难解决。在解决大型问题时，精确的方案往往需要繁重的计算，要完成这些难以处理的计算，必须采用一些近似技术，并构建快速且有可扩展性的系统。

统计推断旨在根据可观察到的事物来了解不可观察到的事物。即，统计推断是基于一个总体或一些样本中的某些观察变量（通常是影响）得出结论的过程，例如关于总体或样本中某些潜在变量（通常是原因）的准时估计、置信区间或区间估计等。

而贝叶斯推理则是从贝叶斯的角度产生统计推断的过程。简而言之，贝叶斯范式是一种统计/概率范式，在这种范式中，每次记录新的观测数据时就会更新由概率分布建模的先验知识，观测数据的不确定性则由另一个概率分布建模。支配贝叶斯范式的整个思想嵌入在所谓的贝叶斯定理中，该定理表达了更新知识（“后验”）、已知知识（“先验”）以及来自观察的知识（“可能性”）之间的关系。

贝叶斯模型选择

贝叶斯定理为

$$P(\Theta_M \mid D, M) = \frac{P(D \mid \Theta_M, M)\, P(\Theta_M \mid M)}{P(D \mid M)}$$

上面的每一项都有一个名称，测量不同的概率:

1. **后验概率:** $P(\Theta_M \mid D, M)$ 是给定数据 D 和具有超参数 Θ_M 的模型 M 的参数值 Θ_M 的条件概率。
2. **可能性:** $P(D \mid \Theta_M, M)$ 是给出模型 (Θ_M, M) 的数据 D 的概率，又称为模型 (Θ_M, M) 的似然。
3. **先验概率:** $P(\Theta_M \mid M)$ 是给定超参数的模型参数的概率，并且在所有可能的数据上被边缘化。

4. **证据:** $P(D \mid M)$ 是给出超参数的数据的概率，并且在给出超参数的所有可能的参数值上被边缘化。

当我们只考虑一个模型时，我们经常忽略模型 M。然而，除非只有一个可能的模型，否则我们仍然需要解决比较可能的模型的元推理问题。虽然这一步通常被称为 模型选择 ，但最好将其看作是 模型比较 ，因为它只能为不同的模型分配相对的概率。

从贝叶斯视角看模型选择是用**概率来表示模型选择的不确定性**。假设要比较模型集 M，模型指的是在观测数据集 D 上的一个概率分布。我们假设**数据是从这些模型中的某一个产生的**，但不确定是哪一个。

假设观测数据为 D，模型 M 的后验概率为

$$P(M \mid D) = \frac{P(D \mid M)\, P(M)}{P(D)} ,$$

其中，分子 $P(D \mid M)$ 是给定 M 的 证据 ，而分母现在是一个 超级证据 ：

$$P(D) = \int P(D \mid M)\, P(M)\, dM$$

在一个离散可数(也许是无限)的模型集的情况下，积分变成一个和

$$P(D) = \sum_k P(D \mid M_k)\, P(M_k) ,$$

每个 模型可能性 $P(D \mid M_k)$ 指的是在 $[0,1]$ 范围内的概率。并引入了模型本身的先验 $P(M)$，它可以看作是整个模型空间上的似然函数，模型参数均被边际化。

$$P(M) = \int P(\Theta_M, M)\, d\Theta_M$$

> 注意这个 模型的贝叶斯定理 与原始贝叶斯定理之间的相似性。

两个实际模型比较的问题:

1. 在不计算证据 $P(D \mid M)$ 的情况下对给定模型进行推理，但这不适用于模型比较。
2. 为了计算 超级证据 ，必须能够指定**所有可能的模型**。

然而，如果只想比较两个可能的模型 M_1 和 M_2，而不指定(或者甚至不知道)可能的模型集，就可以避开第二个问题。此处用 优势比 进行比较：

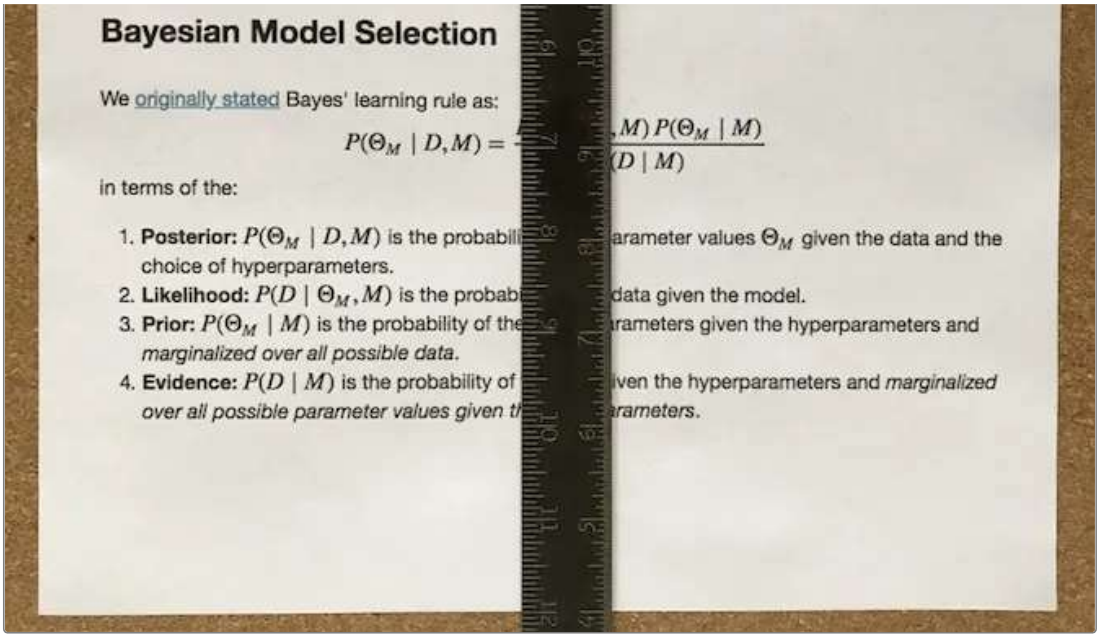
$$\text{odds ratio} = \frac{P(M_1 \mid D)}{P(M_2 \mid D)} = \frac{P(D \mid M_1)\, P(M_1)}{P(D \mid M_2)\, P(M_2)} ,$$

其中， 超级证据 $P(D)$ 在比率中抵消了。右边出现的 模型证据 的比率被称为**贝叶斯因子（Bayes factor）**：

$$\text{Bayes factor} = \frac{P(D \mid M_1)}{P(D \mid M_2)} .$$

举例说明

我们很容易忽视全局概率的计算，所以来看一个简单的例子:



讨论

研究此刻你所观察到的图像 D 在脑海中的两个模型:

- M_1:图像显示一张纸。
- M_2:图像显示两张纸。

这两种模式都是有可能的吗？给出一些为什么不可能是 M_2 的理由。你的论点是否基于先验知识?

这两种模型当然都是可能的，因为尺子可能隐藏了这是并排的两张纸的事实。

然而，实际上不太可能是模型 M_2，因为:

- 这看起来像一张标准的（美国）纸，纸张尺寸作为论点会支持这种可能的结论。
- 退一步讲，如果有两张纸能像这样完美地排成一行，那一定是个惊人的巧合。

第一个论点是基于我们的先验知识，例如:

- 标准（美国）纸的比例是8.5到11。
- 大多数（美国）尺子的单位是英寸。

因此，上面的这种类型的论证出现在模型先验的比率中，即优势比为 $P(M_1)/P(M_2)$。

然而，第二个论点是关于概率的陈述，不依赖于任何先验知识。相反，它出现在上面的贝叶斯因子中。如要了解这是内在机制，我们需要为每个模型定义参数。对于每一张纸，使用四个参数:

- 定义 (x, y) 在纸张的左上角，这可以是图像中的任何位置。
- 纸张的宽度 w 和高度 h，范围覆盖整个图像的宽度和高度。

M_1 和 M_2 的可能性分别是四个和八个参数的积分:

$$P(D \mid M_i) = \int P(D, \Theta_i \mid M_i)\, d\Theta_i$$

对于 M_2，观察到的图像 D 的可能性将为零，除非第二张纸的参数表明它在标尺下完全对齐。因为 M_2 参数范围只是整个图像中的一小部分，所以与 $P(D \mid M_1)$ 相比，$P(D \mid M_2)$ 受到严重的惩罚，从而导致具有较大的Bayes因子。

这是**奥卡姆剃刀**[3]的一个例子：贝叶斯推理倾向于最简单的解释(模型)，独立于任何先验知识。

案例分析:多少峰值?

贝叶斯推理问题通常出现在需要假设概率图模型或根据给定观测值得出模型潜变量的机器学习方法中。本次案例通过先验知识有多少模型来推理选择哪个模型。

定义相关函数

采样方法（MCMC）从由一个因子定义的概率分布中抽取样本。然后，可以从这个分布中得到样本（仅使用未标准化的部分定义），并使用这些样本计算各种准时统计量，如均值和方差，甚至通过密度估计来求得近似分布，从而避免处理涉及后验的棘手计算。

根据采样方法定义如下函数：从两个高斯函数的混合中生成单个特征 x 的一些随机样本。

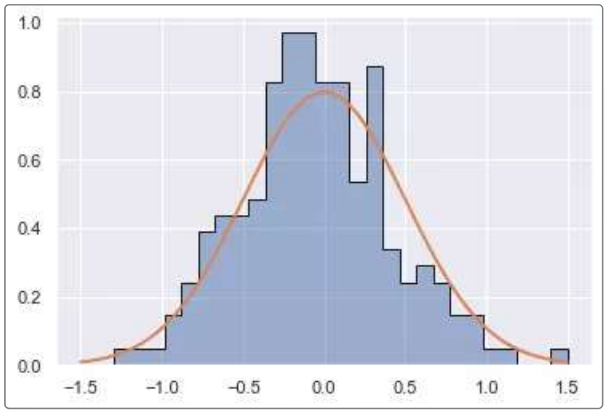
```
def generate_mix(n_total, frac1, mu1, mu2, sigma1, sigma2, seed=123, plot_range=(-1.5, 1.5)):
    gen = np.random.RandomState(seed=seed)
    # 将每个样本分配给其中一个峰值。
    idx = scipy.stats.bernoulli.rvs(1 - frac1, size=n_total, random_state=gen)
    # 设置每个样本的高斯参数。
    mu = np.array([mu1, mu2])[idx]
    sigma = np.array([sigma1, sigma2])[idx]
    # 生成每个样本。
    X = scipy.stats.norm.rvs(mu, sigma, random_state=gen)
    # 可选的绘图.
    if plot_range:
        bins = np.linspace(*plot_range, 30)
        plt.hist(X, bins, histtype='stepfilled', alpha=0.5, density=True)
        plt.hist(X, bins, histtype='step', color='k', lw=1, density=True)
        grid = np.linspace(*plot_range, 201)
        if frac1 > 0:
            pdf1 = scipy.stats.norm.pdf(grid, mu1, sigma1)
            plt.plot(grid, frac1 * pdf1, lw=2)
        if frac1 < 1:
            pdf2 = scipy.stats.norm.pdf(grid, mu2, sigma2)
            plt.plot(grid, (1 - frac1) * pdf2, lw=2)
        plt.show()
    return X
```

生成数据集

生成高斯参数均值为0、方差为0.5，且只有一个峰值的高斯分布。

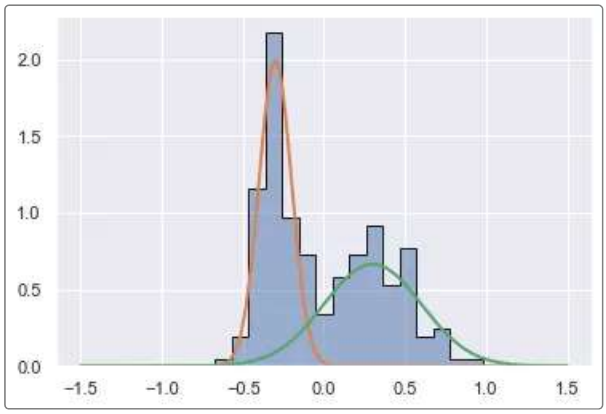
```
n_gen = 200
```

```
Da = generate_mix(n_gen, 1.0, 0.0, np.nan, 0.5, np.nan)
```



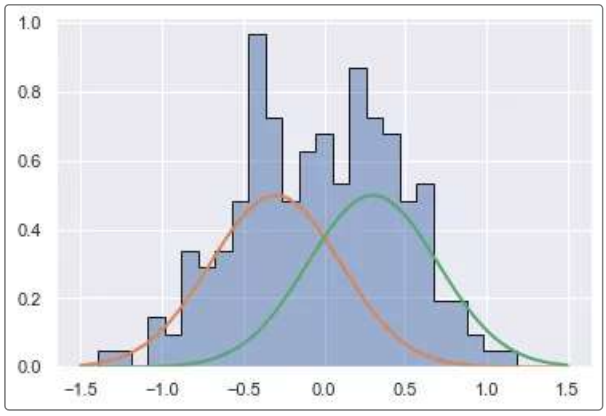
生成第一高斯参数均值为-0.3、方差为0.1，第二高斯参数均值为0.3、方差为0.3，这样就得到具有两个峰值的高斯分布。

```
Db = generate_mix(n_gen, 0.5, -0.3, 0.3, 0.1, 0.3)
```



将上步的方差增大至0.4，其余参数不变，得到如下高斯分布图。

```
Dc = generate_mix(n_gen, 0.5, -0.3, 0.3, 0.4, 0.4)
```



> 注意，每个样本都有从每个高斯分布中提取的整数项，因此只在设置了均值情况下frac1才能真正的实现。

接下来通过比较两个模型来解释这些数据。

- $M1$：单个高斯函数，其平均值 μ 和标准差 σ 未知。
- $M2$：两个等比例高斯函数(`frac1=0.5`)，固定间隔 $\mu_2 - \mu_1 = 0.6$，未知的标准差 σ_1， σ_2。

> 注意，Da 是从 $M1$ 和 Db 中得到的，Dc 是从 $M2$ 中得到的。

讨论

对于这三个数据集，预测 $M1$ 相对于 $M2$ 的 Bayes系数 是否为:

- 远大于1，即强烈倾向于 $M1$
- 约等于1，即仅凭数据无法区分 $M1$ 和 $M2$
- 远小于1，即强烈倾向于 $M2$

解释推理

Da 应该强烈倾向于 $M1$，因为 $M2$ 意味着具有固定的间隔，而现在这种情况表明不可能由 $M2$ 产生。

Db 应该强烈倾向于 $M2$，因为 $M1$ 在合理统计波动范围内是不会重现 $M2$ 的。

Dc 应该不能区分 $M1$ 和 $M2$，因为尽管它是由 $M2$ 产生的，$M1$ 在合理统计波动范围内，是有可能产生这样的数据。

Jeffreys proposed a scale[4] 定义了对于Bayes因子的阈值:

- 大于 10^{+2} 是支持 $M1$ 的“决定性证据”。
- 大于 10^{+1} 是支持 $M1$ 的“有力证据”。
- 小于 10^{-1} 是支持 $M2$ 的“有力证据”。
- 小于 10^{-2} 是支持 $M2$ 的“决定性证据”。

估计模型证据

为了比较给定数据 D 的模型，我们对每个候选模型 M 执行以下步骤:

- 用MCMC进行贝叶斯推理，假设 M，并从后验 $P(\theta \mid D, M)$ 中得到均值和标准差为 (μ, σ) 的样本。
- 使用生成的样本构建后验核密度估计值。
- 使用核密度估计值估算关于 D 在给定 M 时的证据 $P(D \mid M)$。
- 旦我们估算了每个模型的证据，我们就可以计算任意一对模型的贝叶斯因子。
- 最后一步是为每个模型分配相对先验概率，以计算优势比。

> 马尔可夫链蒙特卡罗(Markov Chain Monte Carlo，简称MCMC)，MCMC算法旨在从给定的概率分布中生成样本。

对于每个模型中的参数先验，我们假设 μ 和 $t = log(\sigma)$ 在以下区间上是一致的:

- $M1$: $|\mu| \leq 1$ 和 $0.05 \leq \sigma \leq 1.0$.
- $M2$: $|\mu| \leq 1$ 和 $0.05 \leq \sigma_i \leq 1.0$, with $\mu \equiv (\mu_1 + \mu_2)/2$.

```
mu_range = (-1., +1.)
sigma_range = (0.05, 1.0)
t_range = np.log(sigma_range)
```

下面的证明假设 $M1$ 的代码计算是相当复杂的，其中包含了几个如下几个主题:

- 贝叶斯推理。
- 马尔可夫链蒙特卡洛
- 基于高斯混合模型的核密度估计
- MCMC样本的证据估计

为了建立这个计算，首先定义两个模型的对数后验pdf:

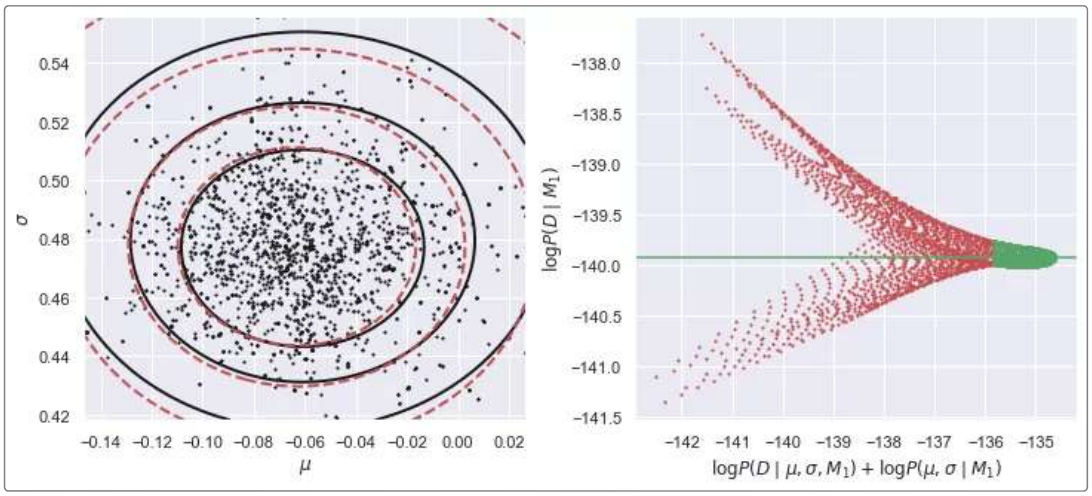
```
def M1_logpost(D, mu, t):
    # 变量变更
    sigma = np.exp(t)
    # 在(mu, sigma)上应用先验概率
    if np.abs(mu) > 1: return -np.inf
    if sigma < 0.05 or sigma > 1.0: return -np.inf
    # 计算并返回可能性函数的对数。
    return scipy.stats.norm.logpdf(D, mu, sigma).sum()


def M2_logpost(D, mu, t1, t2):
    # Perform change of variables.
    mu1 = mu - 0.3
    mu2 = mu + 0.3
    sigma1 = np.exp(t1)
    sigma2 = np.exp(t2)
    # 在(mu, t1, t2)上应用先验概率
    if np.abs(mu) > 1: return -np.inf
    if sigma1 < 0.05 or sigma1 > 1.0: return -np.inf
    if sigma2 < 0.05 or sigma2 > 1.0: return -np.inf
    # 计算并返回可能性函数。
    return np.log(0.5 * (
       scipy.stats.norm.pdf(D, mu1, sigma1) +
       scipy.stats.norm.pdf(D, mu2, sigma2))).sum()
```

现在准备估算观测数据 D 的证据，假设模型 $M1$:

> 函数 `calculate_M1_evidence(D, n_mc=2000, n_grid=50, seed=123)` 主要步骤:
>
> 1. 根据数据为MCMC链选择初始点。
> 2. 使用MCMC从M1后验处生成样本。
> 3. 建立一个参数网格来估计证据。
> 4. 用GMM估计生成的样本的后验密度。
> 5. 评估网格上的密度。
> 6. 使用分子最大的25%网格点进行证据估计。8.用一个图形来总结结果。
>
> 该函数代码略，可联系「数据STUDIO」作者云朵君获取。

估算模型 $M1$ 的观测数据 Da 证据

```
E_Da_M1 = calculate_M1_evidence(Da)
```

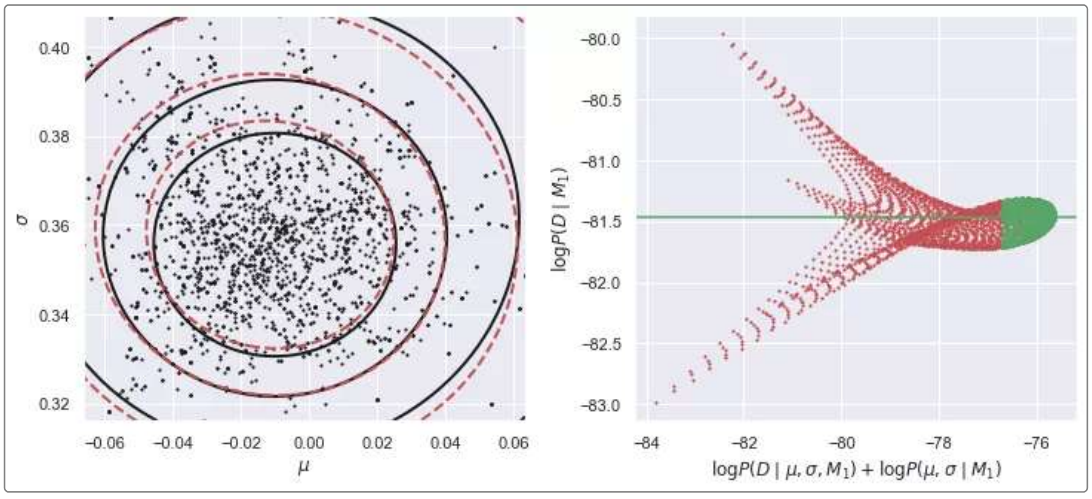


在上面的左图中，黑色实线表示未归一化后验的形状，红色虚等高线表示GMM密度模型与MCMC样本的吻合。两者并不需要完全一致，但一致性越佳将意味着更准确的估计证据 $P(Da \mid M1)$。在这个例子中，我们使用的是单一成分的GMM，但后验分布表现轻微的非高斯分布，所以可以尝试增加MCMC样本的数量，并添加另一个GMM成分。

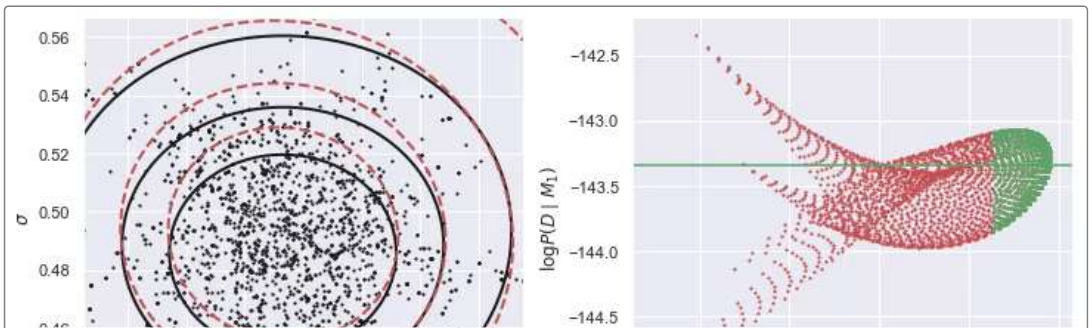
右边的图显示了大量证据的独立估计，计算未归一化后验和GMM密度模型在统一的 $(\mu\ ,\ \sigma)$ 点的二维网格上的比率。为了结合这些独立的估计，取绿色值的中值，因为此处的后验概率是最大的（所以这个过程应该更准确）。查看**MCMC notebook**[5] 给出了一个简单的一维证据估计的例子。

估算模型 $M1$ 的观测数据 Db 和 Dc 的证据

```
E_Db_M1 = calculate_M1_evidence(Db)
```




```
E_Dc_M1 = calculate_M1_evidence(Dc)
```



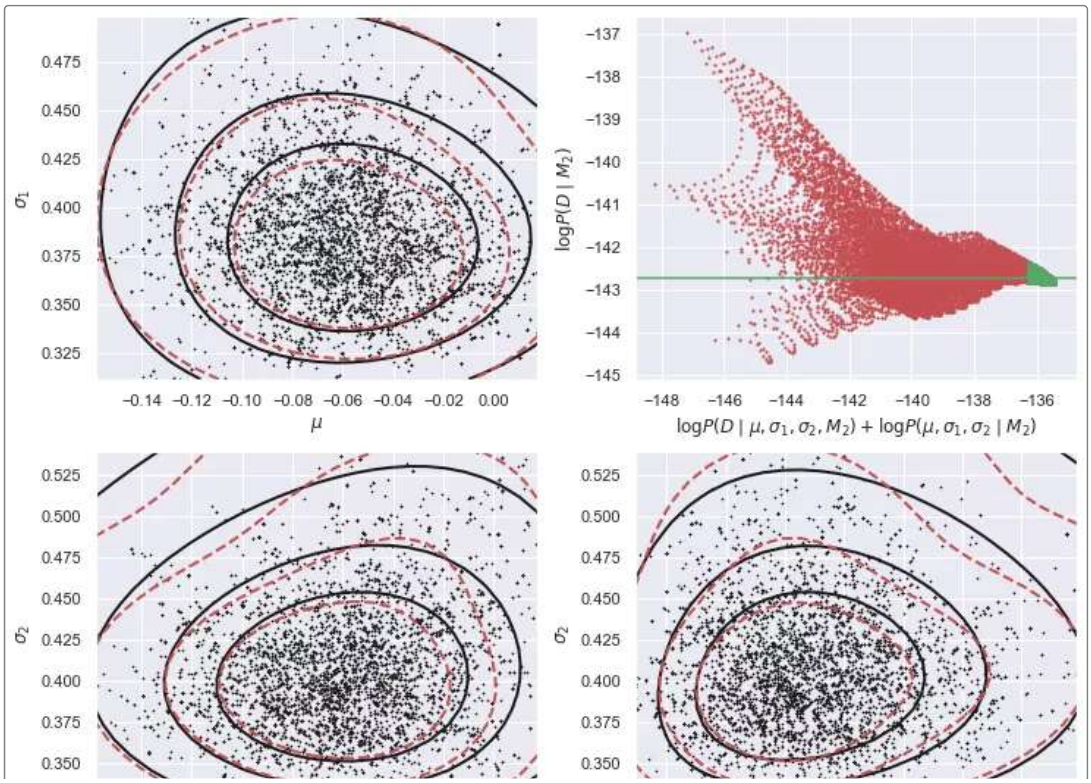
模型 $M2$ 的观测数据 D 的证据

估算模型 $M2$ 的观测数据 D 的证据计算非常相似，但现在在三维参数空间 $(\mu, t1, t2)$:

> 函数 `calculate_M2_evidence(D, n_mc=5000, n_grid=25, seed=123)` 主要步骤:
>
> 1. 根据数据为MCMC链选择起点。
> 2. 使用MCMC从M1后验处生成样本。
> 3. 用sigmai替换ti=log(sigmai)。
> 4. 建立一个参数网格来估计证据。
> 5. 计算网格上的后验分子P(D|mu,sigma1,sigma2) P(mu,sigma1,sigma2)
> 6. 用数值一维积分计算对每一对参数的投影。
> 7. 用GMM估计生成的样本的后验密度。
> 8. 评估网格上的密度。
> 9. 用数值一维积分计算对每一对参数的投影。
> 10. 使用5%的网格点的最大分子的证据估计。
> 11. 用一个图形来总结结果
>
> 该函数代码略，可联系「数据STUDIO」作者云朵君获取。

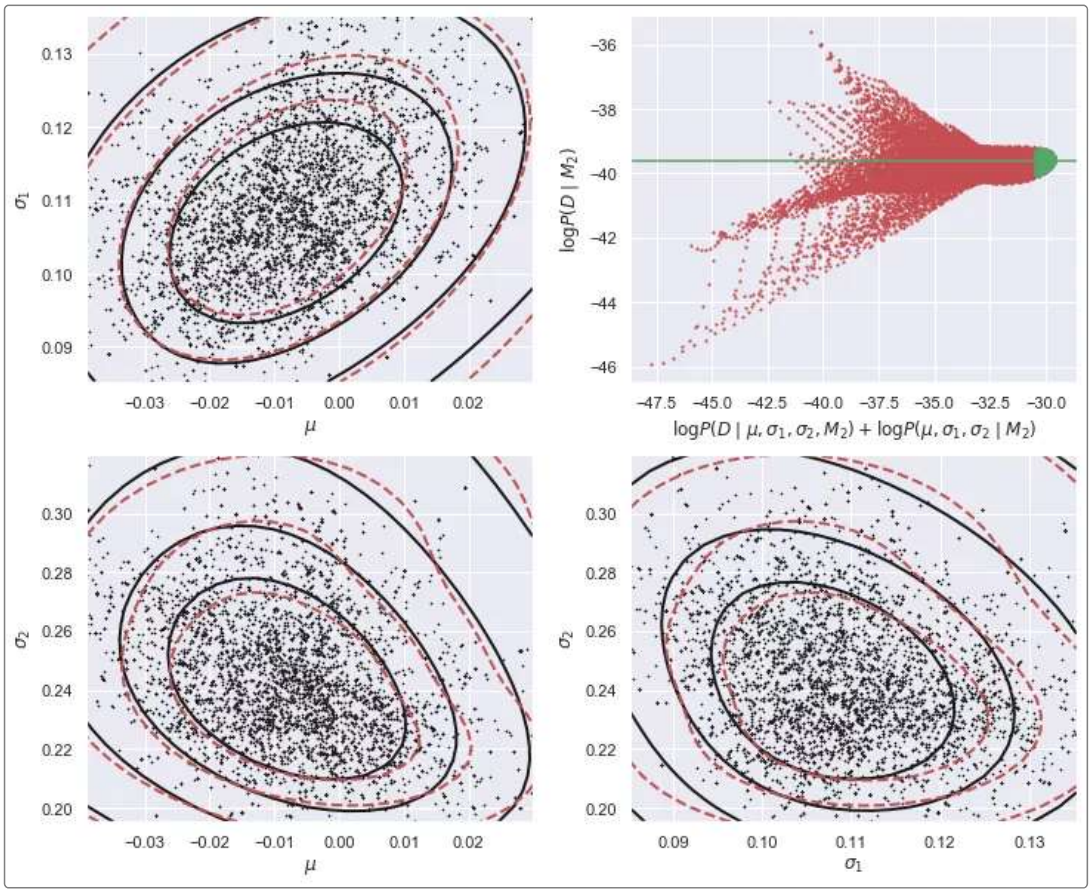
估算模型 $M2$ 的观测数据 Da 证据

```
E_Da_M2 = calculate_M2_evidence(Da)
```



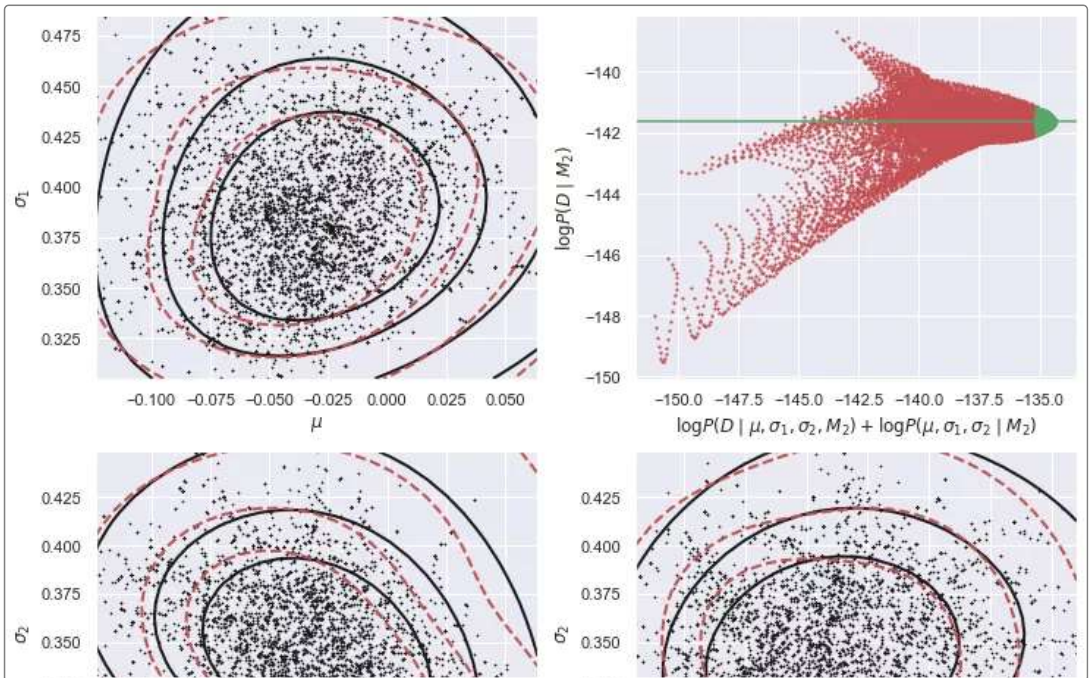
估算模型 $M2$ 的观测数据 Db 证据

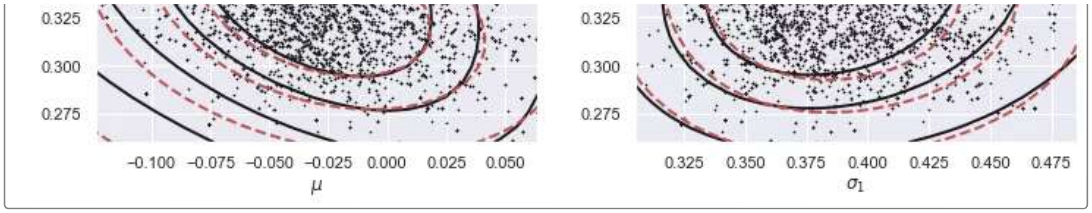
```
E_Db_M2 = calculate_M2_evidence(Db)
```



估算模型 $M2$ 的观测数据 Dc 证据

```
E_Dc_M2 = calculate_M2_evidence(Dc)
```





每一个涉及的计算都提供一个数字，即观测数据 D 在给定模型 M 的证据的对数的估计。

$$\log P(D \mid M)$$

注意，这些都是很小的数字($e^{-100} \simeq 10^{-44}$)，但它们的差异很重要。贝叶斯因子计算如下

$$\text{Bayes' factor} = \exp[\log P(D \mid M_1) - \log P(D \mid M_2)] \ .$$

最后，在没有任何数据的情况下，我们对 $M1$ 和 $M2$ 的相对可能性应用我们的主观先验权重，得到 $M1:M2$ 的优势比:

$$\frac{P(M_1 \mid D)}{P(M_2 \mid D)} = (\text{Bayes' factor}) \times \frac{P(M_1)}{P(M_2)} \ .$$

假设 $M1$ 和 $M2$ 具有相等的先验权值，则优势比等于贝叶斯因子:

```
def summarize(M1_prior=0.5, M2_prior=0.5):
    results = pd.DataFrame({
        'logM1': [E_Da_M1, E_Db_M1, E_Dc_M1],
        'logM2': [E_Da_M2, E_Db_M2, E_Dc_M2]},
        index=('Da', 'Db', 'Dc'))
    results['log10(Bayes)'] = (results['logM1'] - results['logM2']) / np.log(10.)
    results['log10(Odds)'] = results['log10(Bayes)'] + np.log10(M1_prior / M2_prior)
    return results.round(1)

summarize()
```

	logM1	logM2	log10(Bayes)	log10(Odds)
Da	-139.9	-142.7	1.2	1.2
Db	-81.5	-39.6	-18.2	-18.2
Dc	-143.3	-141.6	-0.7	-0.7

简言之:

- 数据 Da(实际上是由 $M1$ 生成的)“强烈”(但不是“决定性”)支持 $M1$。
- 数据 Db(实际上是由 $M2$ 生成的)“决定性”支持 $M2$。
- $M2$ 得到了数据 Dc(由 $M2$ 生成)的支持，但证据并不“有力”。

如果我们有一个先验偏差，即 $M1$ 的可能性是 $M2$ 的10倍，这将主要影响我们对 Dc 的评估，现在结果是它稍微倾向于M1:

```
summarize(M1_prior=10, M2_prior=1)
```

	logM1	logM2	log10(Bayes)	log10(Odds)
Da	-139.9	-142.7	1.2	2.2
Db	-81.5	-39.6	-18.2	-17.2

参考资料

[1] 贝叶斯推理: *https://zhuanlan.zhihu.com/p/75617364*

[2] 贝叶斯模型选择: *https://github.com/dkirkby/MachineLearningStatistics*

[3] 奥卡姆剃刀: *https://en.wikipedia.org/wiki/Occam's_razor*

[4] Jeffreys proposed a scale: *https://en.wikipedia.org/wiki/Bayes_factor#Interpretation*

[5] MCMC notebook: *https://nbviewer.jupyter.org/github/dkirkby/MachineLearningStatistics/blob/master/notebooks/MCMC.ipynb*