吳浩宇

Introduction to Data Analysis 2023. Mar. 07

V. Markov Chain Monte Carlo method.

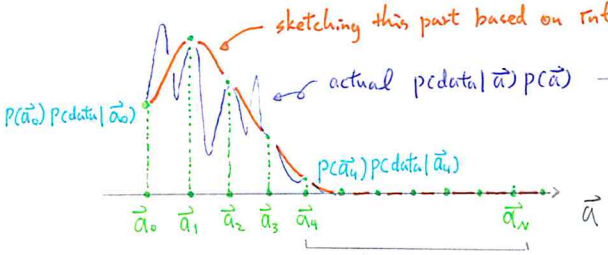
what we want to know (Bayes' theorem):

posterior probability of the model parameter $\vec{a}$: $P(\vec{a}\,|\,data)$

$$\underbrace{P(\vec{a}\,|\,data)}_{\text{posterior probability dist.}} \propto \underbrace{P(data\,|\,\vec{a})}_{\text{likelihood}}\ \overbrace{P(\vec{a})}^{\text{prior probability}} \equiv \underbrace{\pi(\vec{a})}_{\text{unnormalized probability distribution}}$$

We want a numerical method to draw $\pi(\vec{a})$, in particular, when $P(data|\vec{a})$ either has a very complicated analytical form or no analytical fo

approach (i): (Monte Carlo) (quasi) uniform sampling.



Problem 1.
We usually cannot know what is the appropriate density of sampling points $\vec{a}_i$, such that the $P(data|\vec{a})P(\vec{a})$ can be represented with necessary details.

Problem 2.

類比：在處理 χ^2 minimization 時，也只有在接近 χ^2_{min} 處之 $\vec{a}$ 才是重要的

Usually this low-probability region is not important. But in high-dimensional problem (i.e., problems with lots of free parameters), these low-probability samples can cost most of your CPU hours.

approach (ii): Markov Chain Monte Carlo method.

Using this magical method to visit the $\vec{a}$ space iteratively, with a probability that is proportional to $\pi(\vec{a})$

requirement for this method.

(i) ergodic: no region is not unreachable.

(ii) convergence: given any initial distribution of random sampling, we should be able to eventually arrive the desired distribution function $\pi(\vec{a})$

(iii) practicle: it should not be terribly difficult to construct this method, and it should not cost unreasonably large computing power

此方法的好處之一在集中運算資源在描述 $\pi(\vec{a})$ 較大之區域 ($\vec{a}$ 空間中)

吕浩宇

Introduction to Data Analysis 2023. Mar. 07

Markov chain: a sequence of random variable $\vec{x}_i$ where the probability distribution $P(\vec{x}_i)$ only depends on $\vec{x}_{i-1}$, i.e, the variable $\vec{x}_i$ has no memory about what happened "earlier" than $i-1$

Nicholas Constantine Metropolis, 美籍, 芝加哥大学物理学家, 曼哈顿计划最早期参加者之一

A. Metropolis' Principle (earlier 1950s)

If we by to sample $\pi(\vec{a})$ with a Markov chain (i.e., a sequence of random variable $\vec{a}_0, \vec{a}_1, \vec{a}_2, \cdots$ that is "locally correlated"), we will eventually visit everywhere in the $\vec{a}$ space with a density of sampling that is proportional to $\pi(\vec{a})$, as long as the probability to go from one sampling point to the next, $P(\vec{a}_i | \vec{a}_{i-1})$, by construction, satisfy the **detailed balance equation**:

$$\pi(\vec{a}_1) P(\vec{a}_2 | \vec{a}_1) = \pi(\vec{a}_2) P(\vec{a}_1 | \vec{a}_2) \quad \text{——— Eq. (4)}$$

for any $\vec{a}_1$ and $\vec{a}_2$.

物理诠释：位置 $\vec{a}_1$ 之机率密度乘以 $\vec{a}_1$ 到 $\vec{a}_2$ 之跃迁机率等於位置 $\vec{a}_2$ 之机率密度乘以 $\vec{a}_2$ 到 $\vec{a}_1$ 之跃迁机率

(i) A loose argument for the ergodicity (严格之数学证明存在，但必须对机率论中之某定理具有一定的认识)

integrating Eq. (4) with respect to $\vec{a}_1$

$$\Rightarrow \int \pi(\vec{a}_1) P(\vec{a}_2 | \vec{a}_1) \, d\vec{a}_1 = \int \pi(\vec{a}_2) P(\vec{a}_1 | \vec{a}_2) \, d\vec{a}_1$$

$$= \pi(\vec{a}_2) \underbrace{\int P(\vec{a}_1 | \vec{a}_2) \, d\vec{a}_1}_{1.0, \text{ due to probability conservation}} = \pi(\vec{a}_2) \quad \text{——— Eq. (5)}$$

若初始时於 $\vec{a}$ 空间放入許多(無窮多)個 sampling points, 並且其密度正比於 $\pi(\vec{a})$, 且若跟據此初始 sampling points 各別位置, 随机地依 $P(\vec{a}_i | \vec{a}_{i-1})$ 机率函式選取下一批 (無窮多個) sampling points 的位置, 則下一批 sampling points 在 $\vec{a}$ 空间中之密度分布亦等同 $\pi(\vec{a})$。

由此反觀之, $\vec{a}$ 空間中任一點 無論 $\pi(\vec{a})$ 有多低, 都必可由 $\vec{a}$ space 中另一位置之 sampling point 依 $P(\vec{a}_i | \vec{a}_{i-1})$ 随机採樣被尋訪到。不存在不可能被尋訪到之位置。[然必須適當地選取 $P(\vec{a}_i | \vec{a}_{i-1})$ 之到表型式, 使得遍歷整個 $\vec{a}$ 空間不困難 (i.e., 不需要無窮多個 iterations)

(ii) A loose argument for the convergence (類似量子力学, 先看 variable 为不連續之情形之定理, 再 loosely argue 在 variable 为連續之情形定理亦成立)

Defining $P(x_j | x_i) \equiv P_{ij}$ ——— 由於 P_{ij} 为机率, 必然滿足 $\begin{cases} 0 \le P_{ij} \le 1 \\ \sum_j P_{ij} = 1 \end{cases}$

P_{ij}: transition matrix

則由一組 sampling points 之密度分布過渡到下一組 sampling points 之密度分布由 $\overleftrightarrow{P}^T \vec{v}$ 決定, $\vec{v}$ 代表 sampling points 之密度分布 (可想像为 histogram)

定理: $\overleftrightarrow{P}^T$
- must has at least one unity eigenvalue (observing Eq. (5) can see that $\pi(\vec{a})$ is an eigenvector of $\overleftrightarrow{P}^T$ that has unity eigenvalue)
- The absolute values of $\overleftrightarrow{P}^T$ eigenvalues are less than 1.0 (Otherwise, after many operation of $\overleftrightarrow{P}^T$, the number distribution of sampling points will diverge, which is intuitible)

呂浩宇

Introduction to Data Analysis 2023. Mar. 07

若以 $\hat{T}$ 的 eigenfunctions decompose 初始之任意 sampling points 分布 $\vec{a}$，
由 page 9 最底下兩定理可預期經過無窮多次 $\hat{T}$ 的作用之後，僅能留下與 π 平行的 compone
至於要經過多少次 $\hat{T}$ 的作用才能達到好的 convergence 由 $\hat{T}$ 第二大之 eigenvalue 之大小決定.
若 $\hat{T}$ 第二大之 eigenvalue 很小，稱 $\hat{T}$ 為 rapid mixing.

B. How to construct the transition matrix : Metropolis-Hasting Algorithm

(實作見課程網頁 jupyter notebook)

不同的 package 可能採不同的 $q(\vec{a}_2|\vec{a}_1)$ 型式

例如：以 $\vec{a}_1$ 為中心之多維高斯分布

(i) pick a (arbitrary, to some extent...) proposal distribution $q(\vec{a}_2|\vec{a}_1)$

(ii) starting at $\vec{a}_1$, generate a candidate point $\vec{a}_{2c}$ by randomly drawing from the proposal distribution

(iii) evaluate the acceptance probability $\alpha(\vec{a}_1, \vec{a}_{2c})$ that is

$$\alpha(\vec{a}_1, \vec{a}_{2c}) = \min\left(1, \frac{\pi(\vec{a}_{2c})\, q(\vec{a}_1|\vec{a}_{2c})}{\pi(\vec{a}_1)\, q(\vec{a}_{2c}|\vec{a}_1)}\right) \quad \text{——— Eq. (6)}$$

(iv) with the probability $\alpha(\vec{a}_1, \vec{a}_{2c})$, accept the candidate point and set $\vec{a}_2 = \vec{a}_{2c}$.
Otherwise, reject $\vec{a}_{2c}$ and set $\vec{a}_2 = \vec{a}_1$.

注意若 samples 沒被 advanced，即在步驟 (iv) 中 $\vec{a}_{2c}$ 不斷地被 reject，可能代表選取的 $q(\vec{a}_2|\vec{a}_1)$ 型式不理想

→ 此演算法等效地給出 transition probability: $p(\vec{a}_2|\vec{a}_1) = q(\vec{a}_2|\vec{a}_1)\,\alpha(\vec{a}_1, \vec{a}_2)$, $(\vec{a}_2 \neq \vec{a}$

Eq. (6) 左右同乘 $\pi(\vec{a}_1)\, q(\vec{a}_2|\vec{a}_1)$

$$\Rightarrow \pi(\vec{a}_1)\, q(\vec{a}_2|\vec{a}_1)\,\alpha(\vec{a}_1, \vec{a}_2) = \min\left[\pi(\vec{a}_1)\, q(\vec{a}_2|\vec{a}_1),\ \pi(\vec{a}_2)\, q(\vec{a}_2|\vec{a}_1)\right]$$

↓ 左右兩項位置互換

$$= \min\left[\pi(\vec{a}_2)\, q(\vec{a}_1|\vec{a}_2),\ \pi(\vec{a}_1)\, q(\vec{a}_2|\vec{a}_1)\right]$$

observing that

$$\pi(\vec{a}_2)\, q(\vec{a}_1|\vec{a}_2)\,\alpha(\vec{a}_2, \vec{a}_1) = \min\left[\pi(\vec{a}_2)\, q(\vec{a}_1|\vec{a}_2),\ \pi(\vec{a}_1)\, q(\vec{a}_2|\vec{a}_1)\right]$$

$$\Rightarrow \pi(\vec{a}_1)\, q(\vec{a}_2|\vec{a}_1)\,\alpha(\vec{a}_1, \vec{a}_2) = \pi(\vec{a}_2)\, q(\vec{a}_1|\vec{a}_2)\,\alpha(\vec{a}_2, \vec{a}_1)$$

依上述方式定義 transition matrix，可滿足

$$\pi(\vec{a}_1)\, p(\vec{a}_2|\vec{a}_1) = \pi(\vec{a}_2)\, p(\vec{a}_1|\vec{a}_2)$$

↳ ergodicity 及 convergence 理論上不成問題

C. Practical Aspect (以 Python emcee package 實作，見課程網頁 Jupyter notebook)

可同時用多個 samplers，天然地適合平行運算。接近 converge 到 equilibrium state 之前會存過多的 samplers 更於低 $\pi(\vec{a})$ 位置，故必須放棄前 n 步取到的 samplers，n 的大小無法由理論推算，僅能靠對 samplers 的行為之觀察得知。稱此前 n 個 iterations 為 burn-in steps. 把這些 samplers 的分布做成 histogram 即可得到 $p(\vec{a}|\text{data})$，依此說明最有可能之 $\vec{a}$ 為何 (即 best-fit) 與 $\vec{a}$ 之不同 components 的 uncertainties.