

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24 Markov chain Monte Carlo (MCMC) inference

24.1 Introduction

In Chapter 23, we introduced some simple Monte Carlo methods, including rejection sampling and importance sampling. The trouble with these methods is that they do not work well in high dimensional spaces. The most popular method for sampling from high-dimensional distributions is Markov chain Monte Carlo or MCMC. In a survey by SIAM News¹, MCMC was placed in the top 10 most important algorithms of the 20th century. The basic idea behind MCMC is to construct a Markov chain (Section 17.2) on the state space X whose stationary distribution is the target density $p^*(x)$ of interest (this may be a prior or a posterior). That is, we perform a random walk on the state space, in such a way that the fraction of time we spend in each state x is proportional to $p^*(x)$. By drawing (correlated!) samples x_0, x_1, x_2, \dots , from the chain, we can perform Monte Carlo integration wrt p^* . We give the details below.

The MCMC algorithm has an interesting history. It was discovered by physicists working on the atomic bomb at Los Alamos during World War II, and was first published in the open literature in (Metropolis et al. 1953) in a chemistry journal. An extension was published in the statistics literature in (Hastings 1970), but was largely unnoticed. A special case (Gibbs sampling, Section 24.2) was independently invented in 1984 in the context of Ising models and was published in (Geman and Geman 1984). But it was not until (Gelfand and Smith 1990) that the algorithm became well-known to the wider statistical community. Since then it has become wildly popular in Bayesian statistics, and is becoming increasingly popular in machine learning.

It is worth briefly comparing MCMC to variational inference (Chapter 21). The advantages of variational inference are (1) for small to medium problems, it is usually faster; (2) it is deterministic; (3) it is easy to determine when to stop; (4) it often provides a lower bound on the log likelihood. The advantages of sampling

are: (1) it is often easier to implement; (2) it is applicable to a broader range of models, such as models whose size or structure changes depending on the values of certain variables (e.g., as happens in matching problems), or models without nice conjugate priors; (3) sampling can be faster than variational methods when applied to really huge models or datasets.

已思考 14s

译文 (逐句不遗漏)

24 马尔可夫链蒙特卡洛 (MCMC) 推断

24.1 引言

在第23章中，我们介绍了一些简单的蒙特卡洛方法，包括拒绝采样和重要性采样。问题在于，这些方法在高维空间中表现不佳。用于从高维分布中采样的最流行方法是马尔可夫链蒙特卡洛 (MCMC)。在 *SIAM News* 的一次调查中，MCMC 被列为20世纪最重要的10个算法之一。

MCMC 背后的基本思想是：在状态空间 X 上构造一个马尔可夫链（参见第17.2节），使其平稳分布就是我们关心的目标密度 $p^*(x)$ （这可以是先验或后验）。也就是说，我们在状态空间上进行一次随机游走，使得在每个状态 x 上停留的时间比例与 $p^*(x)$ 成正比。通过从该链中抽取（相互相关的！）样本 x_0, x_1, x_2, \dots ，我们可以对相对于 p^* 的积分执行蒙特卡洛估计。下面给出细节。

MCMC 算法有一段有趣的历史。它由在第二次世界大战期间洛斯阿拉莫斯从事原子弹研究的物理学家发现，并首次在公开文献中由 (Metropolis 等, 1953) 在一本化学期刊上发表。此后在统计文献中 (Hastings, 1970) 发表了一个扩展，但基本未被注意到。一个特殊情形 (吉布斯采样，第24.2节) 在1984年于伊辛模型的背景下被独立发明，并由 (Geman 和 Geman, 1984) 发表。但直到 (Gelfand 和 Smith, 1990)，该算法才被更广泛的统计学界所熟知。自那以后，它在贝叶斯统计中变得极其流行，并在机器学习中也越来越受欢迎。

值得将 MCMC 与变分推断 (第21章) 做一个简要比较。变分推断的优点是：(1) 对于小到中等规模的问题，通常更快；(2) 是确定性的；(3) 容易判断何时停止；(4) 它常常为对数似然提供一个下界。采样方法的优点是：(1) 通常更容易实现；(2) 适用范围更广，例如适用于模型的大小或结构会因某些变量的取值而改变（如匹配问题中会发生），或没有漂亮共轭先验的模型；(3) 当应用于特别庞大的模型或数据集时，采样可能比变分方法更快。

通俗解释（一步步来）

1) 为什么要用 MCMC? (高维的尴尬)

- **拒绝采样**: 像在大盒子里随机扔飞镖，只保留落在目标曲线下的点。维度一高，“命中”的概率会急剧变小，几乎全被丢弃 → 浪费巨大。
- **重要性采样**: 从“近似分布”里取样再加权。如果近似分布在高维里和目标分布不贴近，权重会极端不均衡（有的样本权重大到离谱），方差爆炸 → 估计不稳定。
- **MCMC 的思路**: 与其一次性“蒙”对，不如**边走边看**。让一个“随机行走者”在状态空间里游走，并且设计走路规则，使得它**长远来看**在某个区域待的时间正好与目标分布在那里的概率成比例。这样，收集行走者到过的点，就相当于从目标分布取样了。

2) 关键术语用大白话

- **马尔可夫链**: 下一步只取决于“当前在哪”，不关心你之前怎么来的（“只看眼前，不问过往”）。
- **平稳分布**: 如果你让这条链一直走下去，久而久之，落在各个区域的**长期比例**会稳定下来，这个稳定的比例分布就是“平稳分布”。
- **把平稳分布做成我们要的 $p^*(x)$** : 通过设计转移规则（比如 Metropolis-Hastings 的“提议+接受/拒绝”），保证这条链长期逗留的时间分配正好等于目标分布。
- **相关样本**: 链是一步接一步走的，相邻样本会“很像”（相关）。这和独立抽样不同，**但仍然可以用来做蒙特卡洛估计**，只是有效样本数会小些，需要更长的链来平均掉相关性。
- **蒙特卡洛积分**: 想估计 $E_{p^*}[f(X)]$ ，就把链走出来的样本带进去平均：

$$E[f(X)] \approx \frac{1}{N} \sum_{i=1}^N f(x_i)$$

即便样本相关，只要链“混合”得够好、跑得够久，这个平均仍会收敛到真值。

3) 一个直观比喻

把目标分布 $p^*(x)$ 想象成一张“地形图”，高的地方代表概率大、低的地方代表概率小。MCMC 就像一个**会挑地方扎营**的旅人：虽然他随机行动，但规则让他**更愿意在“山谷（高概率）”待久一点、在“山顶（低概率）”待短一点**。你记录他到过的地方及停留时间，就重现了这张地形图的“概率阴影”。

4) 历史脉络 (知道名字不迷路)

- 1953: Metropolis 等提出最初版本（物理/化学背景）。
- 1970: Hastings 推出更一般的扩展（Metropolis-Hastings）。

- 1984: Geman & Geman 在图像上的伊辛模型中独立提出吉布斯采样（是 MH 的一个特例）。
- 1990: Gelfand & Smith 推广到统计界 → 随后在贝叶斯统计与机器学习中大火。

5) 与变分推断 (VI) 的对比, 怎么选?

- VI 的优点
 1. 小/中等规模问题通常更快；
 2. 过程是确定性的（同样输入给同样输出）；
 3. 何时停止好判断（比如看下界收敛）；
 4. 往往给出对数似然的下界（ELBO）。
- 采样（MCMC）的优点
 1. 常常更易实现（不用设计复杂的下界或可积的变分族）；
 2. 适用范围广：
 - 模型结构或维度会随变量取值而变（如匹配/分配等**可变结构**问题）；
 - 没有优雅共轭先验的模型也能用；
 3. 在**特别巨大的**模型或数据集上，有时 MCMC 比 VI 还快（尤其当能构造高效的转移或并行化时）。

小贴士：实际使用里，很多人会把二者结合——先用 VI 给个好初值，再用 MCMC 精修；或用 VI 得到下界/证据近似，用 MCMC 做关键部分的精确抽样检验。

要点总结

- **目标**：从难以直接抽样的高维目标分布 $p^*(x)$ 获得样本，用于估计期望/积分。
- **做法**：构造一条马尔可夫链，让其**平稳分布=目标分布**；沿链收集样本做蒙特卡洛估计。
- **优势**：对复杂、非共轭、甚至**结构可变**的模型普适，常常实现简单；在超大规模问题上有时更快。
- **样本相关**：MCMC 样本彼此相关，但只要“混合”好、运行够久，平均值仍然可靠。
- **与 VI 的取舍**：VI 通常更快、更确定、易停且给下界；MCMC 更通用、实现简、可处理更复杂情形。
- **历史**：Metropolis (1953) → Hastings (1970) → Gibbs (1984) → Gelfand & Smith (1990) 推广普及。

如果你愿意，我可以继续用一个极简的二维例子，演示 MCMC 如何“走路”并收敛到目标分布，以及如何从样本估计期望。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24.2 Gibbs sampling

In this section, we present one of the most popular MCMC algorithms, known as Gibbs sampling.³ (In physics, this method is known as Glauber dynamics or the heat bath method.) This is the MCMC analog of coordinate descent.

24.2.1 Basic idea

The idea behind Gibbs sampling is that we sample each variable in turn, conditioned on the values of all the other variables in the distribution. That is, given a joint sample x_s of all the variables, we generate a new sample x_{s+1} by sampling each component in turn, based on the most recent values of the other variables. For example, if we have $D = 3$ variables, we use

- $x_{s+1,1} \sim p(x_1|x_s, x_2, x_3)$
- $x_{s+1,2} \sim p(x_2|x_s, x_1, x_3)$
- $x_{s+1,3} \sim p(x_3|x_s, x_1, x_2)$

This readily generalizes to D variables. If x_i is a visible variable, we do not sample it, since its value is already known.

The expression $p(x_i|x_{-i})$ is called the full conditional for variable i . In general, x_i may only depend on some of the other variables. If we represent $p(x)$ as a graphical model, we can infer the dependencies by looking at i 's Markov blanket, which are its neighbors in the graph. Thus to sample x_i , we only need to know the values of i 's neighbors. In this sense, Gibbs sampling is a distributed algorithm. However, it is not a parallel algorithm, since the samples must be generated sequentially.

For reasons that we will explain in Section 24.4.1, it is necessary to discard some of the initial samples until the Markov chain has burned in, or entered its stationary distribution. We discuss how to estimate when burnin has occurred in Section 24.4.1. In the examples below, we just discard the initial 25% of the samples, for simplicity.

24.2.2 Example: Gibbs sampling for the Ising model

In Section 21.3.2, we applied mean field to an Ising model.

Here we apply Gibbs sampling.

Gibbs sampling in pairwise MRF/CRF takes the form

$$p(x_t|x_{-t}, \theta) \propto \sum_{s \in nbr(t)} \psi_{st}(x_s, x_t) \quad (24.1)$$

In the case of an Ising model with edge potentials

$\psi_{st} = \exp(J x_s x_t)$, where $x_t \in \{-1, +1\}$, the full conditional becomes

$$p(x_t = +1|x_{-t}, \theta) = \sum_{s \in nbr(t)} \psi_{st}(x_t = +1, x_s) \quad (24.2)$$

$$\begin{aligned} &= \exp[J \sum_{s \in nbr(t)} x_s] \exp[\sum_{s \in nbr(t)} x_s] + \exp[-J \sum_{s \in nbr(t)} x_s] \\ &\quad (24.3) \end{aligned}$$

$$= \exp[J \eta_t] \exp[J \eta_t] + \exp[-J \eta_t] = \text{sigm}(2J \eta_t) \quad (24.4)$$

where J is the coupling strength, $\eta_t = \sum_{s \in nbr(t)} x_s$ and $\text{sigm}(u) = 1/(1 + e^{-u})$ is the sigmoid function. It is easy to see that $\eta_t = x_t(a_t - d_t)$, where a_t is the number of neighbors that agree with (have the same sign as) t , and d_t is the number of neighbors who disagree. If this number is equal, the “forces” on x_t cancel out, so the full conditional is uniform.

We can combine an Ising prior with a local evidence term ψ_t . For example, with a Gaussian observation model, we have $\psi_t(x_t) = N(y_t|x_t, \sigma^2)$. The full conditional becomes

$$\begin{aligned} p(x_t = +1|x_{-t}, y, \theta) &= \exp[J \eta_t] \psi_t(+1) \exp[J \eta_t] \psi_t(+1) + \\ &\quad \exp[-J \eta_t] \psi_t(-1) \quad (24.5) \end{aligned}$$

$$= \text{sigm}(2J \eta_t - \log \psi_t(+1) \psi_t(-1)) \quad (24.6)$$

Now the probability of x_t entering each state is determined both by compatibility with its neighbors (the Ising prior) and compatibility with the data (the local likelihood term).

See Figure 24.1 for an example of this algorithm applied to a simple image denoising problem. The results are similar to mean field (Figure 21.3) except that the final estimate (based on averaging the samples) is somewhat “blurrier”, due to the fact that mean field tends to be over-confident.

24.2.3 Example: Gibbs sampling for inferring the parameters of a GMM

It is straightforward to derive a Gibbs sampling algorithm to “fit” a mixture model, especially if we use conjugate priors. We will focus on the case of mixture of Gaussians, although the results are easily extended to other kinds of mixture models. (The derivation, which follows from the results of Section 4.6, is much easier than the corresponding variational Bayes algorithm in Section

21.6.1.)

Suppose we use a semi-conjugate prior. Then the full joint distribution is given by

$$p(x, z, \mu, \Sigma, \pi) = p(x|z, \mu, \Sigma)p(z|\pi)p(\pi) \prod_{k=1}^K p(\mu_k)p(\Sigma_k) \quad (24.7)$$

$$= N \prod_{i=1}^N \prod_{k=1}^K (\pi_k N(x_i|\mu_k, \Sigma_k)) I(z_i=k) \times \quad (24.8)$$

$$\text{Dir}(\pi|\alpha) \prod_{k=1}^K N(\mu_k|m_0, V_0) IW(\Sigma_k|S_0, v_0) \quad (24.9)$$

We use the same prior for each mixture component. The full conditionals are as follows. For the discrete indicators, we have

$$p(z_i = k|x_i, \mu, \Sigma, \pi) \propto \pi_k N(x_i|\mu_k, \Sigma_k) \quad (24.10)$$

For the mixing weights, we have (using results from Section 3.4)

$$p(\pi|z) = \text{Dir}(\{\alpha_k + N \sum_{i=1}^N I(z_i=k)\} | K, 1) \quad (24.11)$$

For the means, we have (using results from Section 4.6.1)

$$p(\mu_k|\Sigma_k, z, x) = N(\mu_k|m_k, V_k) \quad (24.12)$$

$$V^{-1} k = V^{-1} 0 + N_k \Sigma^{-1} k \quad (24.13)$$

$$m_k = V_k (\Sigma^{-1} k N_k x_k + V^{-1} m_0) \quad (24.14)$$

$$N_k = N \sum_{i=1}^N I(z_i=k) \quad (24.15)$$

$$x_k = N \sum_{i=1}^N I(z_i=k)x_i / N_k \quad (24.16)$$

For the covariances, we have (using results from Section 4.6.2)

$$p(\Sigma_k|\mu_k, z, x) = IW(\Sigma_k|S_k, V_k) \quad (24.17)$$

$$S_k = S_0 + N \sum_{i=1}^N I(z_i=k)(x_i - \mu_k)(x_i - \mu_k)^T \quad (24.18)$$

$$V_k = V_0 + N_k \quad (24.19)$$

See gaussMissingFitGibbs for some Matlab code. (This code can also sample missing values for x , if necessary.)

24.2.3.1 Label switching

Although it is simple to implement, Gibbs sampling for mixture models has a fundamental weakness. The problem is that the parameters of the model θ , and the indicator functions z , are unidentifiable, since we can arbitrarily permute the hidden labels without affecting the likelihood (see Section 11.3.1). Consequently, we cannot just take a Monte Carlo average of the samples to compute posterior means, since what one sample considers the parameters for cluster 1 may be what another sample considers the parameters for cluster 2.

Indeed, if we could average over all modes, we would find $E[\mu_k|D]$ is the same for all k (assuming a symmetric prior). This is called the label switching problem.

This problem does not arise in EM or VBEM, which just "lock on" to a single mode. However, it arises in any method that visits multiple modes. In 1d problems, one

can try to prevent this problem by introducing constraints on the parameters to ensure identifiability, e.g., $\mu_1 < \mu_2 < \mu_3$ (Richardson and Green 1997). However, this does not always work, since the likelihood might overwhelm the prior and cause label switching anyway. Furthermore, this technique does not scale to higher dimensions. Another approach is to post-process the samples by searching for a global label permutation to apply to each sample that minimizes some loss function (Stephens 2000); however, this can be slow.

Perhaps the best solution is simply to “not ask” questions that cannot be uniquely identified. For example, instead of asking for the probability that data point i belongs to cluster k , ask for the probability that data points i and j belong to the same cluster. The latter question is invariant to the labeling. Furthermore, it only refers to observable quantities (are i and j grouped together or not), rather than referring to unobservable quantities, such as latent clusters. This approach has the further advantage that it extends to infinite mixture models, discussed in Section 25.2, where K is unbounded; in such models, the notion of a hidden cluster is not well defined, but the notion of a partitioning of the data is well defined

24.2.4 Collapsed Gibbs sampling

In some cases, we can analytically integrate out some of the unknown quantities, and just sample the rest. This is called a collapsed Gibbs sampler, and it tends to be much more efficient, since it is sampling in a lower dimensional space.

More precisely, suppose we sample z and integrate out θ . Thus the θ parameters do not participate in the Markov chain; consequently we can draw conditionally independent samples $\theta_s \sim p(\theta|z_s, D)$, which will have much lower variance than samples drawn from the joint state space (Liu et al. 1994). This process is called Rao-Blackwellisation, named after the following theorem:

Theorem 24.2.1 (Rao-Blackwell). Let z and θ be dependent random variables, and $f(z, \theta)$ be some scalar function.

Then

$$\text{var}_z, \theta [f(z, \theta)] \geq \text{var}_z [E_\theta [f(z, \theta)|z]] \quad (24.20)$$

This theorem guarantees that the variance of the estimate created by analytically integrating out θ will always be lower (or rather, will never be higher) than the variance of

a direct MC estimate. In collapsed Gibbs, we sample z with θ integrated out; the above Rao-Blackwell theorem still applies in this case (Liu et al. 1994).

We will encounter Rao-Blackwellisation again in Section 23.6. Although it can reduce statistical variance, it is only worth doing if the integrating out can be done quickly, otherwise we will not be able to produce as many samples per second as the naive method. We give an example of this below.

24.2.4.1 Example: collapsed Gibbs for fitting a GMM

Consider a GMM with a fully conjugate prior. In this case we can analytically integrate out the model parameters μ_k , Σ_k and π , and just sample the indicators z . Once we integrate out π , all the z_i nodes become inter-dependent. Similarly, once we integrate out θ_k , all the x_i nodes become inter-dependent, as shown in Figure 24.2(b).

Nevertheless, we can easily compute the full conditionals as follows:

$$p(z_i = k | z_{-i}, x, \alpha, \beta) \propto p(z_i = k | z_{-i}, \alpha, \beta) p(x_i | z_i = k, z_{-i}, \alpha, \beta) \quad (24.21)$$

$$\propto p(z_i = k | z_{-i}, \alpha) p(x_i | x_{-i}, z_i = k, z_{-i}, \beta) p(x_{-i} | z_i = k, z_{-i}, \beta) \quad (24.22)$$

$$\propto p(z_i = k | z_{-i}, \alpha) p(x_i | x_{-i}, z_i = k, z_{-i}, \beta) \quad (24.23)$$

where $\beta = (m_0, V_0, S_0, v_0)$ are the hyper-parameters for the class-conditional densities. The first term can be obtained by integrating out π . Suppose we use a symmetric prior of the form $\pi \sim \text{Dir}(\alpha)$, where $\alpha_k = \alpha/K$. From Equation 5.26 we have

$$p(z_1, \dots, z_N | \alpha) = \Gamma(\alpha) \prod_{k=1}^K \Gamma(N_k + \alpha / K) \Gamma(\alpha / K) \quad (24.24)$$

Hence

$$\begin{aligned} p(z_i = k | z_{-i}, \alpha) &= p(z_1:N | \alpha) p(z_{-i} | \alpha) = \frac{1}{\Gamma(N + \alpha)} \prod_{k=1}^K \frac{\Gamma(N_k + \alpha / K)}{\Gamma(N_k + \alpha / K)} \\ &\times \Gamma(N_k + \alpha / K) \Gamma(N_k - i + \alpha / K) \quad (24.25) \\ &= \frac{\Gamma(N + \alpha - 1)}{\Gamma(N + \alpha)} \prod_{k=1}^K \frac{\Gamma(N_k - i + 1 + \alpha / K)}{\Gamma(N_k - i + \alpha / K)} = \\ &N_k, -i + \alpha / K \quad N + \alpha - 1 \quad (24.26) \end{aligned}$$

where $N_k, -i = n - i$ $I(z_n = k) = N_k - 1$, and where we exploited the fact that $\Gamma(x + 1) = x\Gamma(x)$.

To obtain the second term in Equation 24.23, which is the posterior predictive distribution for x_i given all the other data and all the assignments, we use the fact that

$$p(x_i | x_{-i}, z_{-i}, z_i = k, \beta) = p(x_i | D_{-i}, k) \quad (24.27)$$

where $D_{-i}, k = \{x_j : z_j = k, j = i\}$ is all the data assigned to cluster k except for x_i . If we use a conjugate prior for θ_k ,

we can compute $p(x_i|D-i,k)$ in closed form. Furthermore, we can efficiently update these predictive likelihoods by caching the sufficient statistics for each cluster. To compute the above expression, we remove x_i 's statistics from its current cluster (namely z_i), and then evaluate x_i under each cluster's posterior predictive. Once we have picked a new cluster, we add x_i 's statistics to this new cluster.

Some pseudo-code for one step of the algorithm is shown in Algorithm 1, based on (Sudderth 2006, p94). (We update the nodes in random order to improve the mixing time, as suggested in (Roberts and Sahu 1997).) We can initialize the sample by sequentially sampling from $p(z_i|z_1:i-1, x_1:i)$. (See fmGibbs for some Matlab code, by Yee-Whye Teh.) In the case of GMMs, both the naive sampler and collapsed sampler take $O(NKD)$ time per step.

A comparison of this method with the standard Gibbs sampler is shown in Figure 24.3. The vertical axis is the data log probability at each iteration, computed using $\log p(D|z, \theta) = \sum_{i=1}^N \log [\pi_{zi} p(x_i|\theta_{zi})]$ (24.28)

To compute this quantity using the collapsed sampler, we have to sample $\theta = (\pi, \theta_1:K)$ given the data and the current assignment z .

In Figure 24.3 we see that the collapsed sampler does indeed generally work better than the vanilla sampler. Occasionally, however, both methods can get stuck in poor local modes. (Note that the error bars in Figure 24.3(b) are averaged over starting values, whereas the theorem refers to MC samples in a single run.)

24.2.5 Gibbs sampling for hierarchical GLMs

Often we have data from multiple related sources. If some sources are more reliable and/or data-rich than others, it makes sense to model all the data simultaneously, so as to enable the borrowing of statistical strength. One of the most natural way to solve such problems is to use hierarchical Bayesian modeling, also called multi-level modeling. In Section 9.6, we discussed a way to perform approximate inference in such models using variational methods. Here we discuss how to use Gibbs sampling. To explain the method, consider the following example. Suppose we have data on students in different schools. Such data is naturally modeled in a two-level hierarchy:

we let y_{ij} be the response variable we want to predict for student i in school j . This prediction can be based on school and student specific covariates, x_{ij} . Since the quality of schools varies, we want to use a separate parameter for each school. So our model becomes

$$y_{ij} = x^T \beta_j + \epsilon_{ij} \quad (24.29)$$

We will illustrate this model below, using a dataset from (Hoff 2009, p197), where x_{ij} is the socio-economic status (SES) of student i in school j , and y_{ij} is their math score. We could fit each β_j separately, but this can give poor results if the sample size of a given school is small. This is illustrated in Figure 24.4(a), which plots the least squares regression line estimated separately for each of the $J = 100$ schools. We see that most of the slopes are positive, but there are a few “errant” cases where the slope is negative. It turns out that the lines with extreme slopes tend to be in schools with small sample size, as shown in Figure 24.4(b). Thus we may not necessarily trust these fits.

We can get better results if we construct a hierarchical Bayesian model, in which the β_j are assumed to come from a common prior: $\beta_j \sim N(\mu_w, \Sigma_w)$. This is illustrated in Figure 24.5. In this model, the schools with small sample size borrow statistical strength from the schools with larger sample size, because the β_j 's are correlated via the latent common parents (μ_w, Σ_w) . (It is crucial that these hyper-parameters be inferred from data; if they were fixed constants, the β_j would be conditionally independent, and there would be no information sharing between them.)

To complete the model specification, we must specify priors for the shared parameters. Following (Hoff 2009, p198), we will use the following semi-conjugate forms, for convenience:

$$\mu_w \sim N(\mu_0, V_0) \quad (24.30)$$

$$\Sigma_w \sim IW(\eta_0, S-1) \quad (24.31)$$

$$\sigma^2 \sim IG(v_0/2, v_0\sigma^2_0/2) \quad (24.32)$$

Given this, it is simple to show that the full conditionals needed for Gibbs sampling have the following forms. For the group-specific weights:

$$p(\beta_j | D_j, \Theta) = N(\beta_j | \mu_j, \Sigma_j) \quad (24.33)$$

$$\Sigma_j = (\Sigma_w^{-1} + X_j^T X_j / \sigma^2)^{-1} \quad (24.34)$$

$$\mu_j = \Sigma_j (\Sigma_w^{-1} \mu_w + X_j^T y_j / \sigma^2) \quad (24.35)$$

For the overall mean:

$$p(\mu_w | w_1:J, \Sigma_w) = N(\mu | \mu_N, \Sigma_N) \quad (24.36)$$

$$\Sigma - 1 N = V - 1 0 + J\Sigma - 1 \quad (24.37)$$

$$\mu_N = \Sigma_N(V - 1 0 \mu_0 + J\Sigma - 1 w) \quad (24.38)$$

where $w = 1 J j w_j$. For the overall covariance:

$$p(\Sigma_w | \mu_w, w_1:J) = IW((S_0 + S\mu) - 1, \eta_0 + J) \quad (24.39)$$

$$S\mu = j (w_j - \mu_w)(w_j - \mu_w)^T \quad (24.40)$$

For the noise variance:

$$p(\sigma^2 | D, w_1:J) = IG([v_0 + N]/2, [v_0\sigma^2 0 + SSR(w_1:J)]/2) \quad (24.41)$$

$$SSR(w_1:J) = J \sum_{j=1}^N \sum_{i=1}^{n_j} (y_{ij} - w_i^T j x_{ij})^2 \quad (24.42)$$

Applying Gibbs sampling to our hierarchical model, we get the results shown in Figure 24.4(c). The light gray lines plot the mean of the posterior predictive distribution for each school:

$$E[y_{ij} | x_{ij}] = x_i^T j \hat{w}_j \quad (24.43)$$

where

$$\hat{w}_j = E[w_j | D] \approx \frac{1}{S} \sum_{s=1}^S w(s)_j \quad (24.44)$$

The dark gray line in the middle plots the prediction using the overall mean parameters, $x_i^T j \hat{\mu}_w$. We see that the method has regularized the fits quite nicely, without enforcing too much uniformity. (The amount of shrinkage is controlled by Σ_w , which in turns depends on the hyperparameters; in this example, we used vague values.)

24.2.6 BUGS and JAGS

One reason Gibbs sampling is so popular is that it is possible to design general purpose software that will work for almost any model. This software just needs a model specification, usually in the form a directed graphical model (specified in a file, or created with a graphical user interface), and a library of methods for sampling from different kinds of full conditionals. (This can often be done using adaptive rejection sampling, described in Section 23.3.4.) An example of such a package is BUGS (Lunn et al. 2000), which stands for “Bayesian updating using Gibbs Sampling”. BUGS is very widely used in biostatistics and social science. Another more recent, but very similar, package is JAGS (Plummer 2003), which stands for “Just Another Gibbs Sampler”. This uses a similar model specification language to BUGS.

For example, we can describe the model in Figure 24.5 as follows:

```
model {
  for (i in 1:N) {
```

```

for (j in 1:J) {
  y[i,j] ~ dnorm(y.hat[i,j], tau.y)
  y.hat[i,j] <- inprod(W[j, ], X[i, j, ])
}
}
tau.y <- pow(sigma.y, -2)
sigma.y ~ dunif(0,100)
for (j in 1:J) {
  W[j,] ~ dmnorm(mu, SigmaInv)
}
SigmaInv ~ dwish(S0[,], eta0)
mu ~ dmnorm(mu0, V0inv)
}

```

We can then just pass this model to BUGS or JAGS, which will generate samples for us. See the webpages for details. Although this approach is appealing, unfortunately it can be much slower than using handwritten code, especially for complex models. There has been some work on automatically deriving model-specific optimized inference code (Fischer and Schumann 2003), but fast code still typically requires human expertise.

24.2.7 The Imputation Posterior (IP) algorithm

The Imputation Posterior or IP algorithm (Tanner and Wong 1987) is a special case of Gibbs sampling in which we group the variables into two classes: hidden variables z and parameters θ . This should sound familiar: it is basically an MCMC version of EM, where the E step gets replaced by the I step, and the M step gets replaced the P step. This is an example of a more general strategy called data augmentation, whereby we introduce auxiliary variables in order to simplify the posterior computations (here the computation of $p(\theta|D)$). See (Tanner 1996; van Dyk and Meng 2001) for more information.

24.2.8 Blocking Gibbs sampling

Gibbs sampling can be quite slow, since it only updates one variable at a time (so-called single site updating). If the variables are highly correlated, it will take a long time to move away from the current state. This is illustrated in Figure 24.6, where we illustrate sampling from a 2d Gaussian (see Exercise 24.1 for the details). If the variables are highly correlated, the algorithm will move very slowly through the state space. In particular, the size of the moves is controlled by the variance of the conditional

distributions. If this is in the x_1 direction, and the support of the distribution is L along this dimension, then we need $O((L/ \quad)^2)$ steps to obtain an independent sample.

In some cases we can efficiently sample groups of variables at a time. This is called blocking Gibbs sampling or blocked Gibbs sampling (Jensen et al. 1995; Wilkinson and Yeung 2002), and can make much bigger moves through the state space.

已思考 30s

逐句完整译文（不遗漏）

24.2 吉布斯采样 (Gibbs sampling)

在本节中，我们介绍最常用的 MCMC 算法之一，即吉布斯采样。³（在物理学中，该方法称为 Glauber 动力学或热浴法。）这相当于 MCMC 版的坐标下降。

24.2.1 基本思想

吉布斯采样的思想是：依次对分布中的每个变量进行采样，条件是其他所有变量的当前取值。也就是说，给定一次包含所有变量的联合样本 $x^{(s)}$ ，我们通过依次对各个分量采样来生成一个新样本 $x^{(s+1)}$ ，并且每次采样都以“其他变量的最新值”为条件。例如，如果我们有 $D = 3$ 个变量，我们使用

- $x_1^{(s+1)} \sim p(x_1 | x_2^{(s)}, x_3^{(s)})$
- $x_2^{(s+1)} \sim p(x_2 | x_1^{(s+1)}, x_3^{(s)})$
- $x_3^{(s+1)} \sim p(x_3 | x_1^{(s+1)}, x_2^{(s+1)})$

这可以很容易推广到 D 个变量。如果 x_i 是可观测（可见）变量，我们就不对它采样，因为它的值已知。

表达式 $p(x_i | x_{-i})$ 称为变量 i 的**完整条件分布 (full conditional)**。一般来说， x_i 只可能依赖于部分其他变量。如果我们把 $p(x)$ 表示为一个图模型，我们可以通过观察 i 的**马尔可夫毯**（即图中与它相邻的节点）来推断依赖关系。因此，为了对 x_i 采样，我们只需要知道 i 的邻居的取值。从这个意义上讲，吉布斯采样是一个**分布式**算法。然而它不是并行算法，因为样本必须**顺序**生成。

出于第 24.4.1 节将要解释的原因，有必要丢弃一部分初始样本，直到马尔可夫链已经**烧入 (burn-in)**，即进入它的平稳分布。我们将在第 24.4.1 节讨论如何估计烧入已经发生。在下面的例子中，为了简化，我们直接丢弃初始 25% 的样本。

24.2.2 例子：用于伊辛模型的吉布斯采样

在第 21.3.2 节中，我们把**平均场 (mean field)** 应用到了伊辛模型。这里我们应用吉布斯采样。

在成对 MRF/CRF 中，吉布斯采样具有如下形式

$$p(x_t | x_{-t}, \theta) \propto \prod_{s \in \text{nbr}(t)} \psi_{st}(x_s, x_t) \quad (24.1)$$

在伊辛模型且边势函数为 $\psi(x_s, x_t) = \exp(Jx_s x_t)$ 、并且 $x_t \in \{-1, +1\}$ 的情况下，完整条件变为

$$\begin{aligned} p(x_t = +1 | x_{-t}, \theta) &= \frac{\prod_{s \in \text{nbr}(t)} \psi_{st}(x_t = +1, x_s)}{\prod_{s \in \text{nbr}(t)} \psi_{st}(x_t = -1, x_s)} \\ &\stackrel{(24.2)}{=} \frac{\exp(J \sum_{s \in \text{nbr}(t)} x_s)}{\exp(-J \sum_{s \in \text{nbr}(t)} x_s)} \\ &\stackrel{(24.3)}{=} \frac{\exp(J \eta_t)}{\exp(-J \eta_t)} \\ &\stackrel{(24.4)}{=} \text{sigm}(2J \eta_t) \end{aligned}$$

其中 J 是耦合强度， $\eta_t = \sum_{s \in \text{nbr}(t)} x_s$ ，而 $\text{sigm}(u) = 1/(1 + e^{-u})$ 是 Sigmoid 函数。很容易看出 $\eta_t = x_t(a_t - d_t)$ ，其中 a_t 是与 t 一致（同号）的邻居数量， d_t 是与 t 不一致（异号）的邻居数量。如果这两个数相等，则对 x_t 的“作用力”相互抵消，因此完整条件是均匀的。

我们可以把伊辛先验与一个**局部证据项** ψ_t 结合起来。例如，若观测模型是高斯分布，则 $\psi_t(x_t) = N(y_t | x_t, \sigma^2)$ 。此时完整条件变为

$$p(x_t = +1 | x_{-t}, y, \theta) = \frac{\exp(J\eta_t) \psi_t(+1)}{\exp(J\eta_t) \psi_t(+1) + \exp(-J\eta_t) \psi_t(-1)} \quad (24.5)$$

$$= \text{sigm}(2J\eta_t - \log \frac{\psi_t(+1)}{\psi_t(-1)}) \quad (24.6)$$

现在， x_t 进入每个状态的概率同时由与邻居的相容性（伊辛先验）和与数据的相容性（局部似然项）共同决定。

参见图 24.1，那里给出了该算法应用于一个简单图像去噪问题的示例。结果与平均场（图 21.3）类似，只是最后的估计（基于对样本求平均）要“更模糊”一些，因为平均场往往过于自信。

24.2.3 例子：用吉布斯采样推断 GMM 的参数

如果采用共轭先验，推导一个用于“拟合”混合模型的吉布斯采样算法是直接的。我们将聚焦于**高斯混合模型（GMM）**，尽管结果很容易推广到其他类型的混合模型。（推导基于第 4.6 节的结果，比第 21.6.1 节中的相应变分贝叶斯算法容易得多。）

假设我们使用**半共轭先验**。则完整联合分布为

$$p(x, z, \mu, \Sigma, \pi) = p(x | z, \mu, \Sigma) p(z | \pi) p(\pi) \prod_{k=1}^K p(\mu_k) p(\Sigma_k) \quad (24.7)$$

$$= \prod_{i=1}^N \prod_{k=1}^K (\pi_k N(x_i | \mu_k, \Sigma_k))^{I(z_i=k)} \times \quad (24.8)$$

$$\text{Dir}(\pi | \alpha) \prod_{k=1}^K N(\mu_k | m_0, V_0) \text{IW}(\Sigma_k | S_0, v_0) \quad (24.9)$$

我们对每个混合分量使用相同的先验。完整条件如下。对离散指示变量，有

$$p(z_i = k | x_i, \mu, \Sigma, \pi) \propto \pi_k N(x_i | \mu_k, \Sigma_k) \quad (24.10)$$

对混合权重（用第 3.4 节的结果），有

$$p(\pi | z) = \text{Dir}(\{\alpha_k + \sum_{i=1}^N I(z_i=k)\}_{k=1}^K) \quad (24.11)$$

对均值（用第 4.6.1 节的结果），有

$$p(\mu_k | \Sigma_k, z, x) = N(\mu_k | m_k, V_k) \quad (24.12)$$

$$V_k^{-1} = V_0^{-1} + N_k \Sigma_k^{-1} \quad (24.13)$$

$$m_k = V_k (\Sigma_k^{-1} N_k \bar{x}_k + V_0^{-1} m_0) \quad (24.14)$$

$$N_k = \sum_{i=1}^N I(z_i = k) \quad (24.15)$$

$$\bar{x}_k = \frac{\sum_{i=1}^N I(z_i = k) x_i}{N_k} \quad (24.16)$$

对协方差（用第 4.6.2 节的结果），有

$$p(\Sigma_k | \mu_k, z, x) = \text{IW}(\Sigma_k | S_k, v_k) \quad (24.17)$$

$$S_k = S_0 + \sum_{i=1}^N I(z_i = k) (x_i - \mu_k)(x_i - \mu_k)^\top \quad (24.18)$$

$$v_k = v_0 + N_k \quad (24.19)$$

参见 *gaussMissingFitGibbs* 获取一些 Matlab 代码。（如有需要，这段代码也可以对缺失的 x 值进行采样。）

24.2.3.1 标签置换 (Label switching)

尽管实现简单，用于混合模型的吉布斯采样存在一个根本弱点。问题在于模型参数 θ 和指示变量 z 不可辨识：我们可以任意重排（置换）隐标签而不改变似然（见第 11.3.1 节）。因此，我们不能简单地对样本做蒙特卡洛平均以计算后验均值，因为在某个样本看来属于“簇 1”的参数，可能在另一个样本看来是“簇 2”的参数。事实上，如果我们能对所有模态求平均，会发现 $E[\mu_k | D]$ 对所有 k 都相同（假设先验对称）。这称为**标签置换问题**。

该问题不会出现在 EM 或 VBEM 中，它们只会“锁定”在一个模态。然而，任何会访问多个模态的方法都会遇到该问题。在一维问题中，可以尝试通过对参数施加约束以确保可辨识性来避免此问题，例如 $\mu_1 < \mu_2 < \mu_3$ (Richardson 和 Green, 1997)。但这并不总是有效，因为似然可能压倒先验，依然导致标签置换。此外，这种技巧并不易扩展到高维。另一种方法是对样本进行后处理：为每个样本寻找一个全局标签置换，使某个损失函数最小 (Stephens, 2000)；不过这可能较慢。

或许最佳的解决方案是干脆“不去问”那些不能被唯一识别的问题。例如，不要问“数据点 i 属于簇 k 的概率是多少”，而要问“数据点 i 与 j 属于同一簇的概率是多少”。后者对标签是不变的。此外，它只涉及可观测量 (i 与 j 是否被分到一起)，而不是涉及不可观测的量（如潜在簇）。这种方法还有一个优点：它可以扩展到**无限混合模型**（见第 25.2 节， K 无上界）；在此类模型中，“隐藏簇”的概念并不明确，但对数据进行“划分 (partition)”的概念是明确的。

24.2.4 折叠 (Collapsed) 吉布斯采样

在某些情况下，我们可以**解析地积分消去**部分未知量，只对其余部分采样。这称为**折叠吉布斯采样**，它往往更高效，因为它在更低维的空间中进行采样。

更精确地说，假设我们对 z 采样，并把 θ 积分掉。于是 θ 参数不参与马尔可夫链；因此我们可以抽取条件独立的样本 $\theta^{(s)} \sim p(\theta | z^{(s)}, D)$ ，其方差将远低于从联合状态空间直接抽样得到的方差 (Liu 等, 1994)。这个过程称为**Rao–Blackwell 化**，其名来自下面的定理：

定理 24.2.1 (Rao–Blackwell 定理)。令 z 与 θ 为相关随机变量， $f(z, \theta)$ 为某个标量函数，则

$$\text{var}_{z, \theta}[f(z, \theta)] \geq \text{var}_z[\text{E}_\theta[f(z, \theta) | z]] \quad (24.20)$$

该定理保证：通过解析积分掉 θ 所得到的估计的方差**总是更低**（或者说不会更高）于直接蒙特卡洛估计的方差。在折叠吉布斯中，我们在积分掉 θ 的情况下对 z 采样；上述 Rao–Blackwell 定理在此情形下仍然成立 (Liu 等, 1994)。

我们将在第 23.6 节再次遇到 Rao–Blackwell 化。尽管它可以降低统计方差，**只有当积分能快速完成时**才值得这么做，否则与朴素方法相比，我们每秒能产生的样本更少。我们在下

面给出一个例子。

24.2.4.1 例子：用于拟合 GMM 的折叠吉布斯

考虑一个具有**完全共轭先验**的 GMM。在这种情况下，我们可以解析地积分掉模型参数 μ_k, Σ_k, π ，只对指示变量 z 采样。一旦对 π 做了积分，所有 z_i 节点就会**相互依赖**。类似地，一旦对 θ_k 做了积分，所有 x_i 节点也会**相互依赖**，如图 24.2(b) 所示。尽管如此，我们仍可轻松地计算完整条件如下：

$$p(z_i = k | z_{-i}, x, \alpha, \beta) \propto p(z_i = k | z_{-i}, \alpha, \beta) p(x | z_i = k, z_{-i}, \alpha, \beta) \quad (24.21)$$

$$\propto p(z_i = k | z_{-i}, \alpha) p(x_i | x_{-i}, z_i = k, z_{-i}, \beta) p(x_{-i} | z_i = k, z_{-i}, \beta) \quad (24.22)$$

$$\propto p(z_i = k | z_{-i}, \alpha) p(x_i | x_{-i}, z_i = k, z_{-i}, \beta) \quad (24.23)$$

其中 $\beta = (m_0, V_0, S_0, v_0)$ 是类条件密度的超参数。第一项可以通过积分掉 π 得到。假设我们使用对称先验 $\pi \sim \text{Dir}(\alpha)$ ，其中 $\alpha_k = \alpha/K$ 。由式 (5.26) 可得

$$p(z_1, \dots, z_N | \alpha) = \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)} \prod_{k=1}^K \frac{\Gamma(N_k + \alpha/K)}{\Gamma(\alpha/K)} \quad (24.24)$$

因此

$$p(z_i = k | z_{-i}, \alpha) = \frac{p(z_{1:N} | \alpha)}{p(z_{-i} | \alpha)} = \frac{\frac{1}{\Gamma(N+\alpha)}}{\frac{1}{\Gamma(N+\alpha-1)}} \times \frac{\Gamma(N_k + \alpha/K)}{\Gamma(N_{k,-i} + \alpha/K)} \quad (24.25)$$

$$= \frac{\Gamma(N + \alpha - 1)}{\Gamma(N + \alpha)} \frac{\Gamma(N_{k,-i} + 1 + \alpha/K)}{\Gamma(N_{k,-i} + \alpha/K)} = \frac{N_{k,-i} + \alpha/K}{N + \alpha - 1} \quad (24.26)$$

其中 $N_{k,-i} = \sum_{n \neq i} \mathbf{I}(z_n = k) = N_k - 1$ ，并使用了 $\Gamma(x + 1) = x \Gamma(x)$ 。

为了得到式 (24.23) 中的第二项（给定其他数据及所有赋值时 x_i 的**后验预测分布**），我们利用

$$p(x_i | x_{-i}, z_{-i}, z_i = k, \beta) = p(x_i | D_{-i,k}) \quad (24.27)$$

其中 $D_{-i,k} = \{x_j : z_j = k, j \neq i\}$ 是除 x_i 外当前分到簇 k 的所有数据。如果我们对 θ_k 使用共轭先验，就能以闭式形式计算 $p(x_i | D_{-i,k})$ 。此外，我们可以通过**缓存每个簇的充要统计量**来高效更新这些预测似然。计算上，我们先把 x_i 的统计量从它当前的簇（即 z_i ）中移除，然后在每个簇的后验预测下评估 x_i ；一旦选定了新簇，再把 x_i 的统计量加到该新簇中。

算法 1 给出了该算法一步的伪代码，基于 (Sudderth 2006, 第 94 页)。（按照 (Roberts 和 Sahu 1997) 的建议，我们以**随机顺序**更新各个节点以改善混合时间。）我们可以用按顺序从 $p(z_i | z_{1:i-1}, x_{1:i})$ 采样的方式来**初始化**样本。（参见 *fmGibbs*, Yee-Whye Teh 的一

些 Matlab 代码。) 对于 GMM, 朴素采样器与折叠采样器的每步时间复杂度都是 $O(NKD)$ 。

图 24.3 对该方法与标准吉布斯采样器做了比较。纵轴是每次迭代的数据对数概率, 计算式为

$$\log p(D \mid z, \theta) = \sum_{i=1}^N \log [\pi_{z_i} p(x_i \mid \theta_{z_i})] \quad (24.28)$$

用折叠采样器计算该量时, 我们需要在给定数据及当前赋值 z 的条件下对 $\theta = (\pi, \theta_{1:K})$ 进行采样。

从图 24.3 可见, 折叠采样器的确通常优于普通采样器。不过, 偶尔两种方法都会陷入较差的局部模态。(注意: 图 24.3(b) 中的误差条是在不同起点上取平均的, 而定理谈的是单次运行中的 MC 样本。)

24.2.5 面向分层 GLM 的吉布斯采样

我们经常面对来自多个相关来源的数据。如果某些来源更可靠和/或更“数据丰富”, 那么同时建模所有数据以借力 (borrow strength) 是有意义的。解决此类问题最自然的方法之一是使用分层贝叶斯建模 (也称多层建模)。在第 9.6 节中, 我们讨论了用变分方法对这类模型进行近似推断的方法。这里讨论如何使用吉布斯采样。

用下面的例子来解释方法。假设我们拥有不同学校中学生的数据。这类数据天然适合用两层层级建模: 令 y_{ij} 是我们要为学校 j 中学生 i 预测的响应变量。该预测可以基于学校与学生的特征变量 x_{ij} 。由于学校质量会不同, 我们希望为每所学校使用单独的参数。所以我们的模型是

$$y_{ij} = x_{ij}^\top w_j + \varepsilon_{ij} \quad (24.29)$$

我们将使用 (Hoff 2009, 第 197 页) 的数据集进行演示, 其中 x_{ij} 是学校 j 中学生 i 的社会经济地位 (SES), y_{ij} 是其数学成绩。

我们可以分别拟合每个 w_j , 但如果某个学校的样本量很小, 这会导致较差的结果。这一点由图 24.4(a) 说明: 它对 $J = 100$ 所学校分别绘制了最小二乘回归线。我们看到大多数斜率为正, 但有少数“离群”情况斜率为负。事实证明, 斜率极端的线往往出现在样本量较小的学校, 如图 24.4(b) 所示。因此我们未必信任这些拟合。

如果我们构建一个分层贝叶斯模型, 假设 w_j 来自一个共同先验 $w_j \sim N(\mu_w, \Sigma_w)$, 就能得到更好的结果。图 24.5 给出了示意。在该模型中, 样本量小的学校会通过潜在的共同父节点 (μ_w, Σ_w) 与样本量大的学校共享统计信息, 因为 w_j 之间通过这些超参数相关。(关

键在于这些超参数必须由数据推断；如果把它们固定为常数， w_j 就条件独立，不会发生信息共享。）

为完成模型刻画，我们需要为共享参数指定先验。按照 (Hoff 2009, 第 198 页)，为了方便，我们使用如下半共轭形式：

$$\mu_w \sim N(\mu_0, V_0) \quad (24.30)$$

$$\Sigma_w \sim IW(\eta_0, S_0^{-1}) \quad (24.31)$$

$$\sigma^2 \sim IG(v_0/2, v_0\sigma_0^2/2) \quad (24.32)$$

在此设定下，可较为容易地推出吉布斯采样所需的完整条件具有以下形式。对组别特定权重：

$$p(w_j | D_j, \theta) = N(w_j | \mu_j, \Sigma_j) \quad (24.33)$$

$$\Sigma_j^{-1} = \Sigma^{-1} + X_j^\top X_j / \sigma^2 \quad (24.34)$$

$$\mu_j = \Sigma_j (\Sigma^{-1} \mu + X_j^\top y_j / \sigma^2) \quad (24.35)$$

对总体均值：

$$p(\mu_w | w_{1:J}, \Sigma_w) = N(\mu | \mu_N, \Sigma_N) \quad (24.36)$$

$$\Sigma_N^{-1} = V_0^{-1} + J \Sigma^{-1} \quad (24.37)$$

$$\mu_N = \Sigma_N (V_0^{-1} \mu_0 + J \Sigma^{-1} \bar{w}) \quad (24.38)$$

其中 $\bar{w} = \frac{1}{J} \sum_j w_j$ 。对总体协方差：

$$p(\Sigma_w | \mu_w, w_{1:J}) = IW((S_0 + S_\mu)^{-1}, \eta_0 + J) \quad (24.39)$$

$$S_\mu = \sum_j (w_j - \mu_w)(w_j - \mu_w)^\top \quad (24.40)$$

对噪声方差：

$$p(\sigma^2 | D, w_{1:J}) = IG\left(\frac{v_0 + N}{2}, \frac{v_0\sigma_0^2 + SSR(w_{1:J})}{2}\right) \quad (24.41)$$

$$SSR(w_{1:J}) = \sum_{j=1}^J \sum_{i=1}^{N_j} (y_{ij} - w_j^\top x_{ij})^2 \quad (24.42)$$

把吉布斯采样应用到我们的分层模型，可得到图 24.4(c) 所示结果。浅灰色线为每所学校的后验预测分布均值：

$$\mathbb{E}[y_j \mid x_{ij}] = x_{ij}^\top \hat{w}_j \quad (24.43)$$

其中

$$\hat{w}_j = \mathbb{E}[w_j \mid D] \approx \frac{1}{S} \sum_{s=1}^S w_j^{(s)} \quad (24.44)$$

中间的深灰色线使用总体均值参数 $x_{ij}^\top \hat{\mu}_w$ 进行预测。可以看到，该方法很好地对拟合进行了**正则化**，并未强行让所有学校“过分一致”。（“收缩”的程度由 Σ_w 控制，而它又取决于超参数；在这个例子里，我们使用了“宽松（vague）”的取值。）

24.2.6 BUGS 与 JAGS

吉布斯采样之所以如此流行，一个原因在于可以设计**通用软件**，几乎适用于任何模型。这类软件只需要一个模型说明（通常以有向图模型的形式，在文件中指定或用图形界面创建），以及一个**按不同完整条件采样的方法库**。（这通常可用第 23.3.4 节介绍的**自适应拒绝采样**完成。）此类软件的一个例子是 BUGS (Lunn 等, 2000)，其名称来自“Bayesian Updating Using Gibbs Sampling”。BUGS 在生物统计与社会科学中被广泛使用。另一个较新但非常类似的软件是 JAGS (Plummer, 2003)，全称“Just Another Gibbs Sampler”。它使用与 BUGS 类似的模型说明语言。

例如，我们可以如下描述图 24.5 中的模型：

```
model {
  for (i in 1:N) {
    for (j in 1:J) {
      y[i,j] ~ dnorm(y.hat[i,j], tau.y)
      y.hat[i,j] <- inprod(W[j, ], X[i, j, ])
    }
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0, 100)
  for (j in 1:J) {
    W[j, ] ~ dmnorm(mu, SigmaInv)
  }
  SigmaInv ~ dwish(S0[,], eta0)
  mu ~ dnorm(mu0, V0inv)
}
```

然后我们只需把该模型交给 BUGS 或 JAGS，它就会为我们生成样本。详见各自网页。

尽管这种方法很吸引人，但不幸的是，相比于**手写代码**，它在复杂模型上可能慢得多。已经有一些自动派生“模型特定的、优化的推断代码”的工作 (Fischer 和 Schumann,

2003），但要得到快速代码通常仍然需要人类专家参与。

24.2.7 充填-后验 (Imputation Posterior, IP) 算法

IP 算法 (Tanner 和 Wong, 1987) 是吉布斯采样的一个特例，我们把变量分成两类：隐藏变量 z 与参数 θ 。这听上去很熟悉：它基本上是 EM 的一个 MCMC 版本，其中 E 步被 **I步 (Imputation, 充填)** 取代，M 步被 **P步 (Posterior, 后验)** 取代。这是“数据增广 (data augmentation)”这一更一般策略的一个例子，即我们引入**辅助变量**来简化后验计算 (这里是计算 $p(\theta | D)$)。更多信息参见 (Tanner, 1996; van Dyk 和 Meng, 2001)。

24.2.8 分组 (Blocking) 吉布斯采样

吉布斯采样可能相当慢，因为它一次只更新一个变量 (所谓**单点更新**)。如果变量之间高度相关，算法需要很长时间才能从当前状态“挪动”出去。图 24.6 对此做了说明：我们在其中展示了从二维高斯分布采样 (细节见练习 24.1)。当变量高度相关时，算法会非常缓慢地穿过状态空间。特别地，“步子大小”由**条件分布的方差**控制。如果方差沿 x_1 方向，而该方向上分布的有效“跨度”是 L ，那么我们需要 $O((L/)^2)$ 步才能得到一个独立样本。(原式处留有空白符号，表示由条件方差决定的特征尺度，通常可理解为条件标准差。)

在某些情况下，我们可以一次高效地采样一组变量。这称为**分组 (或块) 吉布斯采样** (Jensen 等, 1995; Wilkinson 和 Yeung, 2002)，它能在状态空间中迈出大得多的步伐。

通俗易懂的讲解

1) 吉布斯采样到底干了啥？

把所有变量排成一列：第 1 个变量、第 2 个变量、……第 D 个变量。每一轮我们从头到尾走一遍：

- 用“别的变量的最新值”做条件，给第 1 个变量抽一个新值；
- 再用“刚更新过的第 1 个 + 还没更新的其他”做条件，给第 2 个变量抽一个新值；
- ……直到最后一个。

这一遍走完，得到一个新的“全体变量”的样本。不断重复，就得到一串样本序列。只要设计合理，长远来看，这串样本会像是从目标分布里抽出来的。

2) “完整条件分布”“马尔可夫毯”是啥？

- **完整条件** $p(x_i | x_{-i})$ ：在“其他都给定”的前提下，第 i 个变量的分布。
- 在图模型里，第 i 个变量通常只依赖它的**邻居** (马尔可夫毯)，因此采样 x_i 时只要看它的邻居即可。这让吉布斯**局部化**、实现简单。
- 但要注意：更新是**串行的** (上一部的结果会影响下一步)，所以不是并行算法。

3) 为什么要丢掉前面 25% (烧入) ?

一开始我们往往从一个“随便”的初值出发，链条需要时间“忘记初值”、进入“长期稳定”的状态（平稳分布）。这段前期就叫**烧入**。丢掉它，更稳妥。实际工作中会用诊断工具来判断是否烧入充分。

4) 伊辛模型例子里为什么会出现 Sigmoid?

- 邻居跟你同号越多 (a_t 大、 η_t 大)，就越“鼓励”你取 +1；反之越“鼓励”取 -1。
- 把这种“鼓励”写成指数形式、再标准化，恰好变成了 Sigmoid：
 $\Pr(x_t = +1 \mid \cdot) = \text{sigm}(2J\eta_t)$ 。
- 如果再有观测数据（局部似然 ψ_t ），就把“邻居的劝说”和“数据的证据”加在一起：式 (24.6) 里 $-\log \frac{\psi_t(+1)}{\psi_t(-1)}$ 就是“数据站哪边”的证据量。

5) GMM 参数的吉布斯采样怎么做？

- 轮流做三类事情：
 - 给簇标签 z_i** ：按 $\pi_k N(x_i \mid \mu_k, \Sigma_k)$ 的相对大小抽一个 k 。
 - 给混合权重 π** ：用 Dirichlet (把计数加到超参数)。
 - 给每个簇的 μ_k, Σ_k** ：用正态-逆 Wishart 的后验公式更新 (式 24.12–24.19)。
- 就这样“贴标签—更换参数—再贴标签—再更换参数……”循环。

6) 为什么会有“标签置换”问题？

- GMM 的簇 1、簇 2……只是**名字**，调换名字不影响模型。
- 因此不同样本里“谁是 1、谁是 2”可能不一致，直接对参数逐簇求平均会变得毫无意义。
- 解决思路：要么加“有序约束”（一维还能试试，高维很难），要么后处理统一标签（代价大），要么干脆问“与标签无关”的问题——例如“两个点在不在同一簇”。

7) 折叠吉布斯 & Rao–Blackwell：为什么更稳？

- 如果能把 θ **积分掉**，只对 z 采样，再在需要的时候从 $p(\theta \mid z)$ 抽样，就能减少随机波动。
- Rao–Blackwell 定理保证：这样的估计**方差更小**（或不更大）。
- 代价：要能**快速积分**；否则时间都花在算积分上，得不偿失。

8) 折叠 GMM 具体咋做？

- 把 π, μ_k, Σ_k 都积分掉，只对 z_i 抽样。
- $p(z_i = k \mid \cdot)$ 分成两块：
 - 先验计数**： $\frac{N_{k,-i} + \alpha/K}{N + \alpha - 1}$ (Dirichlet–多项式的结果)；
 - 后验预测**： $p(x_i \mid D_{-i,k})$ (用共轭得到闭式，靠簇内充要统计量就能快速更新)。
- 这样常常混合更快、方差更小。

9) 分层 GLM 的吉布斯采样 ("借力使力")

- 每个学校一条回归线 $y_{ij} = x_{ij}^\top w_j + \varepsilon$ 。
- 如果某学校样本少，单独拟合很不稳。办法：假设 w_j 来自同一正态先验 $N(\mu_w, \Sigma_w)$ 。
- 这会让“样本少的学校”从“样本多的学校”**借到统计信息**，从而得到更稳健的估计（**层级收缩**）。式 (24.33–24.42) 给出了所有完整条件，直接轮流采样就好。

10) BUGS/JAGS：写模型就能跑吉布斯

- 你把概率模型用它们的语言写出来（示例代码已给出），软件就会自动给出抽样器。
- 缺点：通用性强、**速度可能慢**；复杂模型想要快，往往还是得写专门的高效代码。

11) IP 算法：MCMC 版 EM

- 把变量分成“隐藏 z ”和“参数 θ ”，交替做：
 - **I步**：按当前 θ 给 z 充填样本；
 - **P步**：按当前 z 给 θ 的后验抽样。
- 这是“数据增广”的一个实例：加辅助变量把难题分解成容易的条件更新。

12) 为什么要“分组 (Blocking) ”更新？

- 单点更新在变量高度相关时会“挪不动”，要很多步才像是一次独立抽样。
- 如果能一次给一组强相关变量**联合采样**，就相当于迈更大的步子，**混合更快**。

要点总结

- **吉布斯采样**：依次从各变量的**完整条件分布**采样，形成一条马尔可夫链；只需看每个变量的**马尔可夫毯**。
- **烧入**：丢弃前期样本以消除初值影响；实践中常用诊断判断是否充分烧入。
- **伊辛模型**：完整条件化为 Sigmoid（邻居一致性 + 数据证据共同决定）；做图像去噪时直观有效。
- **GMM 参数推断**：轮流采样 z, π, μ_k, Σ_k ；注意**标签置换**，应提问“与标签无关”的量（如共簇概率）。
- **折叠吉布斯 & Rao–Blackwell**：把参数积分掉、只采样关键变量，能显著**降方差**、常**更高效**（前提是积分快）。
- **分层 GLM**：让组参数 w_j 共享先验 $N(\mu_w, \Sigma_w)$ ，实现**信息借力与收缩**，提高小样本组的稳健性。
- **BUGS/JAGS**：通用吉布斯平台，易用但可能较慢；复杂模型仍常需手写优化。
- **IP 算法**：EM 的 MCMC 版本（I步 + P步），属于数据增广思路。

- **块更新**: 一次联合更新一组相关变量，可大幅加快混合、减少相关性。

如果你想，我可以用一个小型的二维示例（含代码与图）演示：标准吉布斯 vs. 块吉布斯/折叠吉布斯在混合速度上的差别，以及怎么做后验预测。

这一节主要在讲什么？我不太懂

已思考 9s

下面用大白话把这整节在说啥捋清楚，抓住主线就不晕啦 ↴

一句话概括

这节讲的是**吉布斯采样** (Gibbs sampling)：一种非常常用的 MCMC 方法。它的做法是按顺序、一个接一个地给每个变量重新抽值（在“其他变量取当前值”的前提下），不断循环，就能得到来自目标分布的样本，用来看做贝叶斯推断、积分、预测等。

它解决的是什么问题？

很多模型的“整体分布”太复杂，**一次性联合采样**几乎不可能。但“单个变量在其他变量给定时的条件分布”往往**简单又可抽样**。吉布斯采样就把“大问题”拆成一串“小条件分布”来抽。

核心做法（算法怎么走）

简单版流程：

1. **初始化**所有变量的一个状态 $x^{(0)}$ （随便给也行）。
2. **一轮更新**（第 s 轮 → 第 $s+1$ 轮）：
 - 抽 $x_1 \sim p(x_1 | x_2, \dots, x_D)$ （用“其他变量的最新值”为条件）
 - 抽 $x_2 \sim p(x_2 | x_1, \dots, x_D)$
 - ...直到抽完 x_D 。得到新状态 $x^{(s+1)}$ 。
3. 重复很多轮，**丢掉前期的“烧入（burn-in）”样本**（比如前 25%），后面的样本就近似来自目标分布了（虽然相邻样本有相关性）。

小提示：在图模型里，抽 x_i 时通常**只用看它的“邻居”（马尔可夫毯）**，实现很省事。

直观比喻

像一屋人合租做决定：每次只让**一位室友**在“其他人当下意见”的前提下更新自己的意见；绕屋一圈大家都更新一次，就算一轮。轮得足够多，整体意见的分布就稳定到我们想要的目标分布。

这节的几个关键例子/主题

1. 伊辛模型 (图像去噪)

抽像素 x_t 时, 只看邻居像素的取值和观测证据; “邻居越多同号, 越倾向跟他们一边”。数学上会出现 Sigmoid 概率 (很自然)。

2. 高斯混合模型 (GMM) 参数学习

轮流:

- 给每个样本的簇标签 z_i 重新抽 (看当前参数);
- 给混合权重 π 、每个簇的均值 μ_k 和协方差 Σ_k 重新抽 (看当前分配)。

但会遇到**标签置换**: 簇 1/2/3 的“名字”可以互换, 导致“逐簇求平均的后验参数”没意义。解决思路是**问与标签无关的问题**, 比如“两个点落同一簇的概率”。

3. 折叠 (Collapsed) 吉布斯

如果能把一些参数 (如 π, μ, Σ) **积分掉**, 只对标签 z 抽, 维度变低、**方差更小、更高效** (Rao-Blackwell 定理保证“更稳”)。但前提是这些积分**算得快**。

4. 分组/块 (Blocking) 吉布斯

单个变量更新在**强相关**时走得很慢; 若能**成组联合抽样**, 就像迈大步, 混合会快很多。

5. 层级 GLM (多学校回归)

给每个学校参数 w_j , 再让所有 w_j 共享一个上一层的先验 (μ_w, Σ_w) 。样本少的学校能从样本多的学校**借力**, 得到稳健拟合。吉布斯采样可直接按闭式完整条件轮流更新。

6. 工具: BUGS / JAGS

把模型用它们的语言写出来, 软件就能自动跑吉布斯; **易用但可能慢**, 复杂模型想要快通常还得手写优化代码。

7. IP 算法 (Imputation-Posterior)

把变量分成“隐藏 z ”和“参数 θ ”, 交替给 z 做“充填(I)”和给 θ 做“后验(P)”抽样; 可把它看作**EM 的 MCMC 版**, 属于“数据增广”思路。

为什么这样做是对的?

在合理条件下, 这个“轮流按完整条件抽样”的马尔可夫链, 会把**平稳分布**收敛到目标后验分布。因此, 足够长之后的样本 (去掉烧入) 就能拿来做期望、预测等蒙特卡洛估计。

这节你需要带走的要点

- **吉布斯采样 = 按完整条件, 一个个变量轮换抽样**; 实现常很简单, 因为只需能从每个条件分布取样。
- **烧入要丢**: 前期样本受初值影响大, 要丢掉。
- **伊辛模型**: 邻居影响 + 观测证据 \rightarrow 抽样概率常是 Sigmoid。
- **GMM**: 可行但有**标签置换**; 多用**标签不变**的量 (如同簇概率) 来回答问题。

- **折叠吉布斯**: 把参数积分掉、只采样关键离散变量，**更稳更快**（如果积分便宜）。
- **块更新**: 强相关时联合更新一组变量，**加速混合**。
- **层级模型**: 通过共享先验实现**信息借力/收缩**，吉布斯更新有闭式。
- **BUGS/JAGS**: 写模型即可自动抽样；灵活但可能慢。
- **IP/数据增广**: EM 思路的 MCMC 化，交替对隐藏和参数抽样。

如果你愿意，我可以用一个**极简的小例子（含数字）**演示吉布斯采样怎么一步步“转起来”，以及折叠/块更新各自带来的加速感受。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24.3 Metropolis Hastings algorithm Although Gibbs sampling is simple, it is somewhat restricted in the set of models to which it can be applied. For example, it is not much help in computing $p(w|D)$ for a logistic regression model, since the corresponding graphical model has no useful Markov structure. In addition, Gibbs sampling can be quite slow, as we mentioned above. Fortunately, there is a more general algorithm that can be used, known as the Metropolis Hastings or MH algorithm, which we describe below.

24.3.1 Basic idea

The basic idea in MH is that at each step, we propose to move from the current state x to a new state x' with probability $q(x'|x)$, where q is called the proposal distribution (also called the kernel). The user is free to use any kind of proposal they want, subject to some conditions which we explain below. This makes MH quite a flexible method. A commonly used proposal is a symmetric Gaussian distribution centered on the current state, $q(x'|x) = N(x'|x, \Sigma)$; this is called a random walk Metropolis algorithm. We discuss how to choose Σ in Section 24.3.3. If we use a proposal of the form $q(x'|x) = q(x'|x)$, where the new state is independent of the old state, we get a method known as the independence sampler, which is similar to importance sampling (Section 23.4).

Having proposed a move to x' , we then decide

whether to accept this proposal or not according to some formula, which ensures that the fraction of time spent in each state is proportional to $p^*(x)$. If the proposal is accepted, the new state is x' , otherwise the new state is the same as the current state, x (i.e., we repeat the sample).

If the proposal is symmetric, so $q(x'|x) = q(x|x')$, the acceptance probability is given by the following formula:
 $r = \min(1, p^*(x')/p^*(x))$ (24.45)

We see that if x is more probable than x' , we definitely move there (since $p^*(x')/p^*(x) > 1$), but if x is less probable, we may still move there anyway, depending on the relative probabilities. So instead of greedily moving to only more probable states, we occasionally allow "downhill" moves to less probable states. In Section 24.3.6, we prove that this procedure ensures that the fraction of time we spend in each state x is proportional to $p^*(x)$.

If the proposal is asymmetric, so $q(x'|x) \neq q(x|x')$, we need the Hastings correction, given by the following:
 $r = \min(1, \alpha)$ (24.46)

$$\alpha = p^*(x')q(x|x')/p^*(x)q(x|x) = p^*(x')/q(x|x')p^*(x)/q(x|x) \quad (24.47)$$

This correction is needed to compensate for the fact that the proposal distribution itself (rather than just the target distribution) might favor certain states.

An important reason why MH is a useful algorithm is that, when evaluating α , we only need to know the target density up to a normalization constant. In particular, suppose $p^*(x) = 1/Z \tilde{p}(x)$, where $\tilde{p}(x)$ is an unnormalized distribution and Z is the normalization constant. Then
 $\alpha = (\tilde{p}(x')/Z) q(x|x') / (\tilde{p}(x)/Z) q(x|x)$ (24.48)

so the Z 's cancel. Hence we can sample from p^* even if Z is unknown. In particular, all we have to do is evaluate \tilde{p} pointwise, where $\tilde{p}(x) = p^*(x)Z$.

24.3.2 Gibbs sampling is a special case of MH

It turns out that Gibbs sampling, which we discussed in Section 24.2, is a special case of MH. In particular, it is equivalent to using MH with a sequence of proposals of the form

$$q(x_i|x_{-i}) = p(x_i|x_{-i})I(x_i = x_{-i}) \quad (24.49)$$

That is, we move to a new state where x_i is sampled from its full conditional, but x_{-i} is left unchanged.

We now prove that the acceptance rate of each such proposal is 1, so the overall algorithm also has an acceptance rate of 100%. We have

$$\begin{aligned}\alpha &= p(x_i)q(x_{-i} | x_i) / p(x_{-i})q(x_i | x_{-i}) = p(x_i | x_{-i})p(x_{-i})p(x_i | x_{-i}) \\ &\quad - p(x_i | x_{-i})p(x_{-i})p(x_i | x_{-i}) \quad (24.50) \\ &= p(x_i | x_{-i})p(x_{-i})p(x_i | x_{-i}) / p(x_i | x_{-i})p(x_{-i})p(x_i | x_{-i}) = 1\end{aligned}\quad (24.51)$$

where we exploited the fact that $x_{-i} = x_{-i}$, and that $q(x_i | x_{-i}) = p(x_i | x_{-i})$.

The fact that the acceptance rate is 100% does not necessarily mean that Gibbs will converge rapidly, since it only updates one coordinate at a time (see Section 24.2.8). Fortunately, there are many other kinds of proposals we can use, as we discuss below.

24.3.3 Proposal distributions

For a given target distribution p^* , a proposal distribution q is valid or admissible if it gives a non-zero probability of moving to the states that have non-zero probability in the target. Formally, we can write this as

$$\text{supp}(p^*) \subseteq \cup_x \text{supp}(q(\cdot | x)) \quad (24.52)$$

For example, a Gaussian random walk proposal has non-zero probability density on the entire state space, and hence is a valid proposal for any continuous state space. Of course, in practice, it is important that the proposal spread its probability mass in just the right way. Figure 24.7 shows an example where we use MH to sample from a mixture of two 1D Gaussians using a random walk proposal, $q(x_{-i} | x_i) = N(x_{-i} | x_i, v)$. This is a somewhat tricky target distribution, since it consists of two well separated modes. It is very important to set the variance of the proposal v correctly: If the variance is too low, the chain will only explore one of the modes, as shown in Figure 24.7(a), but if the variance is too large, most of the moves will be rejected, and the chain will be very sticky, i.e., it will stay in the same state for a long time. This is evident from the long stretches of repeated values in Figure 24.7(b). If we set the proposal's variance just right, we get the trace in Figure 24.7(c), where the samples clearly explore the support of the target distribution. We discuss how to tune the proposal below.

One big advantage of Gibbs sampling is that one does not need to choose the proposal distribution, and furthermore, the acceptance rate is 100%. Of course, a

100% acceptance can trivially be achieved by using a proposal with variance 0 (assuming we start at a mode), but this is obviously not exploring the posterior. So having a high acceptance is not the ultimate goal. We can increase the amount of exploration by increasing the variance of the Gaussian kernel. Often one experiments with different parameters until the acceptance rate is between 25% and 40%, which theory suggests is optimal, at least for Gaussian target distributions. These short initial runs, used to tune the proposal, are called pilot runs.

24.3.3.1 Gaussian proposals

If we have a continuous state space, the Hessian H at a local mode \hat{w} can be used to define the covariance of a Gaussian proposal distribution. This approach has the advantage that the Hessian models the local curvature and length scales of each dimension; this approach therefore avoids some of the slow mixing behavior of Gibbs sampling shown in Figure 24.6.

There are two obvious approaches: (1) an independence proposal, $q(w | w) = N(w | \hat{w}, H^{-1})$ or (2), a random walk proposal, $q(w | w) = N(w | w, s^2 H^{-1})$, where s^2 is a scale factor chosen to facilitate rapid mixing. (Roberts and Rosenthal 2001) prove that, if the posterior is Gaussian, the asymptotically optimal value is to use $s^2 = 2.382/D$, where D is the dimensionality of w ; this results in an acceptance rate of 0.234.

For example, consider MH for binary logistic regression. From Equation 8.7, we have that the Hessian of the log-likelihood is $H = XTX$, where $D = \text{diag}(\mu_i(1 - \mu_i))$ and $\mu_i = \text{sigm}(\hat{w}^T x_i)$. If we assume a Gaussian prior, $p(w) = N(0, V_0)$, we have $H = V^{-1} 0 + H$, so the asymptotically optimal Gaussian proposal has the form

$$q(w | w) = N(w, 2.382 D V^{-1} 0 + XTX - 1) \quad (24.53)$$

See (Gamerman 1997; Rossi et al. 2006; Fruhwirth-Schnatter and Fruhwirth 2010) for further details. The approach is illustrated in Figure 24.8, where we sample parameters from a 1d logistic regression model fit to some SAT data. We initialize the chain at the mode, computed using IRLS, and then use the above random walk Metropolis sampler.

If you cannot afford to compute the mode or its Hessian XDX , an alternative approach, suggested in (Scott 2009), is

to approximate the above proposal as follows:

$$q(w \mid w) = N(w, V - 1) + 6\pi^2 XTX^{-1} \quad (24.54)$$

24.3.3.2 Mixture proposals

If one doesn't know what kind of proposal to use, one can try a mixture proposal, which is a convex combination of base proposals:

$$q(x \mid x) = \sum_{k=1}^K w_k q_k(x \mid x) \quad (24.55)$$

where w_k are the mixing weights. As long as each q_k is individually valid, the overall proposal will also be valid.

24.3.3.3 Data-driven MCMC The most efficient proposals depend not just on the previous hidden state, but also the visible data, i.e., they have the form $q(x \mid x, D)$. This is called data-driven MCMC (see e.g., (Tu and Zhu 2002)). To create such proposals, one can sample (x, D) pairs from the forwards model and then train a discriminative classifier to predict $p(x|f(D))$, where $f(D)$ are some features extracted from the visible data.

Typically x is a high-dimensional vector (e.g., position and orientation of all the limbs of a person in a visual object detector), so it is hard to predict the entire state vector, $p(x|f(D))$. Instead we might train a discriminative detector to predict parts of the state-space, $p(x_k|f_k(D))$, such as the location of just the face of a person. We can then use a proposal of the form

$$q(x \mid x, D) = \pi_0 q_0(x \mid x) + \sum_{k=1}^K \pi_k q_k(x_k \mid f_k(D)) \quad (24.56)$$

where q_0 is a standard data-independent proposal (e.g., random walk), and q_k updates the k 'th component of the state space. For added efficiency, the discriminative proposals should suggest joint changes to multiple variables, but this is often hard to do.

The overall procedure is a form of generate and test: the discriminative proposals $q(x \mid x)$ generate new hypotheses, which are then "tested" by computing the posterior ratio $p(x \mid D) / p(x \mid D')$, to see if the new hypothesis is better or worse. By adding an annealing step, one can modify the algorithm to find posterior modes; this is called simulated annealing, and is described in Section 24.6.1. One advantage of using the mode-seeking version of the algorithm is that we do not need to ensure the proposal distribution is reversible.

24.3.4 Adaptive MCMC

One can change the parameters of the proposal as the algorithm is running to increase efficiency. This is called

adaptive MCMC. This allows one to start with a broad covariance (say), allowing large moves through the space until a mode is found, followed by a narrowing of the covariance to ensure careful exploration of the region around the mode.

However, one must be careful not to violate the Markov property; thus the parameters of the proposal should not depend on the entire history of the chain. It turns out that a sufficient condition to ensure this is that the adaption is “faded out” gradually over time. See e.g., (Andrieu and Thoms 2008) for details.

24.3.5 Initialization and mode hopping

It is necessary to start MCMC in an initial state that has non-zero probability. If the model has deterministic constraints, finding such a legal configuration may be a hard problem in itself. It is therefore common to initialize MCMC methods at a local mode, found using an optimizer.

In some domains (especially with discrete state spaces), it is a more effective use of computation time to perform multiple restarts of an optimizer, and to average over these modes, rather than exploring similar points around a local mode. However, in continuous state spaces, the mode contains negligible volume (Section 5.2.1.3), so it is necessary to locally explore around each mode, in order to visit enough posterior probability mass.

24.3.6 Why MH works

To prove that the MH procedure generates samples from p^* , we have to use a bit of Markov chain theory, so be sure to read Section 17.2.3 first.

The MH algorithm defines a Markov chain with the following transition matrix:

$$p(x' | x) = q(x' | x)r(x | x') \text{ if } x' = x \\ q(x | x) + (1 - r(x | x)) \text{ otherwise} \quad (24.57)$$

This follows from a case analysis: if you move to x' from x , you must have proposed it (with probability $q(x' | x)$) and it must have been accepted (with probability $r(x' | x)$); otherwise you stay in state x , either because that is what you proposed (with probability $q(x | x)$), or because you proposed something else (with probability $q(x' | x)$) but it was rejected (with probability $1 - r(x' | x)$).

Let us analyse this Markov chain. Recall from Section 17.2.3.4 that a chain satisfies detailed balance if

$$p(x_{\cdot} | x)p^*(x) = p(x|x_{\cdot})p^*(x_{\cdot}) \quad (24.58)$$

We also showed that if a chain satisfies detailed balance, then p^* is its stationary distribution. Our goal is to show that the MH algorithm defines a transition function that satisfies detailed balance and hence that p^* is its stationary distribution. (If Equation 24.58 holds, we say that p^* is an invariant distribution wrt the Markov transition kernel q .)

Theorem 24.3.1. If the transition matrix defined by the MH algorithm (given by Equation 24.57) is ergodic and irreducible, then p^* is its unique limiting distribution.

Proof. Consider two states x and x_{\cdot} . Either

$$p^*(x)q(x_{\cdot} | x) < p^*(x_{\cdot})q(x|x_{\cdot}) \quad (24.59)$$

or

$$p^*(x)q(x_{\cdot} | x) > p^*(x_{\cdot})q(x|x_{\cdot}) \quad (24.60)$$

We will ignore ties (which occur with probability zero for continuous distributions). Without loss of generality, assume that $p^*(x)q(x_{\cdot} | x) > p^*(x_{\cdot})q(x|x_{\cdot})$. Hence

$$\alpha(x_{\cdot} | x) = p^*(x_{\cdot})q(x|x_{\cdot}) / p^*(x)q(x|x_{\cdot}) < 1 \quad (24.61)$$

Hence we have $r(x_{\cdot} | x) = \alpha(x_{\cdot} | x)$ and $r(x|x_{\cdot}) = 1$.

Now to move from x to x_{\cdot} we must first propose x_{\cdot} and then accept it. Hence

$$p(x_{\cdot} | x) = q(x_{\cdot} | x)r(x_{\cdot} | x) = q(x_{\cdot} | x)p^*(x_{\cdot})q(x|x_{\cdot})$$

$$p^*(x)q(x_{\cdot} | x) = p^*(x_{\cdot})p^*(x)q(x|x_{\cdot}) \quad (24.62)$$

Hence

$$p^*(x)p(x_{\cdot} | x) = p^*(x_{\cdot})q(x|x_{\cdot}) \quad (24.63)$$

The backwards probability is

$$p(x|x_{\cdot}) = q(x|x_{\cdot})r(x|x_{\cdot}) = q(x|x_{\cdot}) \quad (24.64)$$

since $r(x|x_{\cdot}) = 1$. Inserting this into Equation 24.63 we get

$$p^*(x)p(x_{\cdot} | x) = p^*(x_{\cdot})p(x|x_{\cdot}) \quad (24.65)$$

so detailed balance holds wrt p^* . Hence, from Theorem 17.2.3, p^* is a stationary distribution. Furthermore, from Theorem 17.2.2, this distribution is unique, since the chain is ergodic and irreducible.

24.3.7 Reversible jump (trans-dimensional) MCMC

Suppose we have a set of models with different numbers of parameters, e.g., mixture models in which the number of mixture components is unknown. Let the model be denoted by m , and let its unknowns (e.g., parameters) be denoted by $x_m \in X_m$ (e.g., $X_m = R_m$, where n_m is the dimensionality of model m). Sampling in spaces of differing dimensionality is called transdimensional MCMC

(Green 2003). We could sample the model indicator $m \in \{1, \dots, M\}$ and sample all the parameters from the product space M $m=1 \times M_m$, but this is very inefficient. It is more parsimonious to sample in the union space $X = \cup_{m=1}^M \{m\} \times M_m$, where we only worry about parameters for the currently active model.

The difficulty with this approach arises when we move between models of different dimensionality. The trouble is that when we compute the MH acceptance ratio, we are comparing densities defined in different dimensionality spaces, which is meaningless. It is like trying to compare a sphere with a circle. The solution, proposed by (Green 1998) and known as reversible jump MCMC or RJMCMC, is to augment the low dimensional space with extra random variables so that the two spaces have a common measure.

Unfortunately, we do not have space to go into details here. Suffice it to say that the method can be made to work in theory, although it is a bit tricky in practice. If, however, the continuous parameters can be integrated out (resulting in a method called collapsed RJMCMC), much of the difficulty goes away, since we are just left with a discrete state space, where there is no need to worry about change of measure. For example, (Denison et al. 2002) includes many examples of applications of collapsed RJMCMC applied to Bayesian inference from adaptive basis-function models. They sample basis functions from a fixed set of candidates (e.g., centered on the data points), and integrate out the other parameters analytically. This provides a Bayesian alternative to using RVMs or SVMs.

已思考 31s

下面我先给出逐句完整译文（不中断、不遗漏），再用通俗解释把难点讲清，最后总结要点方便复习。

逐句完整译文（不遗漏）

24.3 Metropolis–Hastings 算法

虽然吉布斯采样很简单，但它可应用的模型集合有些受限。比如，对于逻辑回归模型来

说，计算 $p(\mathbf{w} | D)$ 时它并没有什么帮助，因为相应的图模型没有有用的马尔可夫结构。此外，正如上面提到的，吉布斯采样可能会相当慢。

幸运的是，有一种更通用的算法可以使用，即 Metropolis–Hastings (MH) 算法，我们会在下文描述。

24.3.1 基本思想

MH 的基本思想是：在每一步，我们以概率 $q(x' | x)$ 从当前状态 x 提议 (propose) 移动到一个新状态 x' ，其中 q 称为**提议分布**（也叫核/内核，kernel）。用户可以自由选择任何类型的提议，只要满足我们下文说明的一些条件。这让 MH 十分灵活。一个常用的提议是以当前状态为中心的对称高斯分布， $q(x' | x) = N(x' | x, \Sigma)$ ；这称为**随机游走 Metropolis 算法**。我们在 24.3.3 节讨论如何选择 Σ 。如果我们使用形如 $q(x' | x) = q(x')$ 的提议（即新状态与旧状态独立），就得到一种称为**独立采样器 (independence sampler) **的方法，它与重要性采样 (23.4 节) 相似。

在提出移动到 x' 之后，我们根据某个公式来决定是否接受这个提议，从而确保在每个状态上花费的时间比例与 $p^*(x)$ 成正比。如果提议被接受，新状态就是 x' ；否则新状态与当前状态相同，仍为 x （也就是说，重复该样本）。

如果提议是**对称的**，即 $q(x' | x) = q(x | x')$ ，则接受概率由下式给出：

$$r = \min(1, \frac{p^*(x')}{p^*(x)}) \quad (24.45)$$

可以看到，如果 x' 比 x 更可能（因为 $\frac{p^*(x')}{p^*(x)} > 1$ ），我们一定会移动过去；但即便 x' 更不可能，我们也有时仍会过去，取决于相对概率。因此，与其只“贪心地”向更可能的状态移动，我们也会偶尔允许“下坡”移动到较不可能的状态。在 24.3.6 节，我们将证明该过程能保证在每个状态 x 上花费的时间比例与 $p^*(x)$ 成正比。

如果提议是**非对称的**，即 $q(x' | x) \neq q(x | x')$ ，我们需要 Hastings 校正，给出如下：

$$r = \min(1, \alpha) \quad (24.46)$$

$$\alpha = \frac{p^*(x') q(x | x')}{p^*(x) q(x' | x)} = \frac{p^*(x')/q(x' | x)}{p^*(x)/q(x | x')} \quad (24.47)$$

该校正用来补偿这样一个事实：可能是**提议分布本身**（而不仅仅是目标分布）在偏好某些状态。

MH 是一种有用算法的一个重要原因在于：在计算 α 时，我们只需要知道目标密度到一个**归一化常数**为止。具体地，假设 $p^*(x) = \frac{1}{Z} \tilde{p}(x)$ ，其中 $\tilde{p}(x)$ 是**未归一化**分布， Z 是归一化常数。那么

$$\alpha = \frac{(\tilde{p}(x')/Z) q(x | x')}{(\tilde{p}(x)/Z) q(x' | x)} \quad (24.48)$$

因此 Z 会相互抵消。于是即便 Z 未知，我们也能从 $p^{(*)}$ 采样。特别地，我们只需逐点评估 \tilde{p} ，其中 $\tilde{p}(x) = p^{(*)}(x) Z$ 。

24.3.2 吉布斯采样是 MH 的一个特例

事实证明，我们在 24.2 节讨论的吉布斯采样是 MH 的一个特例。具体说，它等价于使用一系列如下形式的 MH 提议：

$$q(x' | x) = p(x_i | x_{-i}) I(x'_i = x_i) \quad (24.49)$$

也就是说，我们移动到一个新状态，其中 x_i 按其**完整条件分布**采样，而 x_{-i} 保持不变。

我们现在证明，这样的每个提议的**接受率都是 1**，因此整体算法的接受率为 100%。有

$$\alpha = \frac{p(x') q(x | x')}{p(x) q(x' | x)} = \frac{p(x'_i | x_{-i}) p(x_{-i}) p(x_i | x_{-i})}{p(x_i | x_{-i}) p(x_{-i}) p(x'_i | x_{-i})} = (24.50-24.51)$$

这里我们利用了 $x'_i = x_i$ ，以及 $q(x' | x) = p(x_i | x_{-i})$ 这一事实。

接受率 100% 并不必然意味着吉布斯会快速收敛，因为它一次只更新一个坐标（见 24.2.8 节）。幸运的是，我们还可以使用很多其他类型的提议，下面将讨论。

24.3.3 提议分布

对于给定的目标分布 $p^{(*)}$ ，若一个提议分布 q 能以非零概率移动到目标中**非零概率**的那些状态，则称其是**有效/可受理的**。形式化地，可写为

$$\text{supp}(p^{(*)}) \subseteq \bigcup_x \text{supp}(q(\cdot | x)) \quad (24.52)$$

例如，高斯随机游走提议在整个状态空间上都有非零概率密度，因此对于任何连续状态空间都是有效提议。

当然，在实践中，重要的是提议要以“正好合适的方式”分布它的概率质量。图 24.7 展示了一个例子：我们用 MH 从**两个一维高斯的混合**中采样，并使用随机游走提议 $q(x' | x) = N(x' | x, v)$ 。这是一个有些棘手的目标分布，因为它包含两个分离良好的模态。**正确设定提议的方差 v 非常重要**：如果方差太小，链只会探索其中一个模态（见图 24.7(a)）；而如果方差太大，大多数移动会被拒绝，链会非常“粘”（sticky），即在同一状态停留很久——这在图 24.7(b) 的一长串重复值中很明显。如果把方差设得恰到好处，就得到图 24.7(c) 的轨迹，样本显然覆盖了目标分布的支撑。下面我们会讨论如何调参。

吉布斯采样的一个巨大优点是：**不用选择提议分布**，而且接受率是 100%。当然，把提议的方差设为 0（假设我们从某个模态开始）也能“轻松”达到 100% 的接受率，但显然这并未

探索后验。所以高接受率并非最终目标。我们可以通过增大高斯核的方差来增加探索量。实践中常通过试验不同参数，使接受率落在 25%–40% 之间；理论表明这在高斯型目标分布下是较优的。这些用于调提议的短初跑称为试运行（pilot runs）。

24.3.3.1 高斯提议

若状态空间是连续的，可用局部模态 $\hat{\mathbf{w}}$ 处的 Hessian H 来定义高斯提议分布的协方差。好处是 Hessian 刻画了每个维度的局部曲率与长度尺度；因此可避免图 24.6 中吉布斯采样的某些慢混合行为。

有两种显而易见的方法：(1) 独立提议， $q(\mathbf{w}' \mid \mathbf{w}) = N(\mathbf{w}' \mid \hat{\mathbf{w}}, H^{-1})$ ；或 (2) 随机游走提议， $q(\mathbf{w}' \mid \mathbf{w}) = N(\mathbf{w}' \mid \mathbf{w}, s^2 H^{-1})$ ，其中 s^2 是为加快混合而选的尺度因子。

(Roberts and Rosenthal, 2001) 证明：若后验为高斯，渐近最优的取值是

$$s^2 = \frac{2.38^2}{D},$$

其中 D 是 \mathbf{w} 的维度；这会带来大约 0.234 的接受率。

例如，考虑对二元逻辑回归做 MH。由式 (8.7) 知，对数似然的 Hessian 为 $H_\ell = X^\top D X$ ，其中 $D = \text{diag}(\mu_i(1 - \mu_i))$ ， $\mu_i = \text{sigm}(\hat{\mathbf{w}}^\top \mathbf{x}_i)$ 。若先验为高斯 $p(\mathbf{w}) = N(0, V_0)$ ，则

$$H = V_0^{-1} + H_\ell = V_0^{-1} + X^\top D X,$$

于是渐近最优的高斯随机游走提议形如

$$q(\mathbf{w}' \mid \mathbf{w}) = N(\mathbf{w}', \frac{2.38^2}{D} (V_0^{-1} + X^\top D X)^{-1}) \quad (24.53)$$

参见 (Gamerman 1997; Rossi 等 2006; Frühwirth-Schnatter 和 Frühwirth 2010) 了解更多细节。该方法在图 24.8 中有所展示：我们对某个 SAT 数据拟合一维逻辑回归，先用 IRLS 求模态初始化链，再用上述随机游走 Metropolis 采样器。

如果计算模态或其 Hessian $X^\top D X$ 代价太高，(Scott 2009) 建议如下近似该提议：

$$q(\mathbf{w}' \mid \mathbf{w}) = N(\mathbf{w}', (V_0^{-1} + \frac{6}{\pi^2} X^\top X)^{-1}) \quad (24.54)$$

24.3.3.2 混合提议

如果不知道用哪类提议，可以尝试混合提议，即若干基础提议的凸组合：

$$q(x' \mid x) = \sum_{k=1}^K w_k q_k(x' \mid x) \quad (24.55)$$

其中 w_k 是混合权重。只要每个 q_k 各自有效，整体提议也会有效。

24.3.3.3 数据驱动的 MCMC

最高效的提议不仅依赖前一时刻的隐藏状态，还依赖可观测数据，即形如 $q(x' | x, D)$ 。这称为**数据驱动的 MCMC**（见如 Tu & Zhu, 2002）。要构造此类提议，可以从前向模型中采样 (x, D) 对，然后训练判别式分类器来预测 $p(x | f(D))$ ，其中 $f(D)$ 是从可见数据中提取的特征。

通常 x 是高维向量（如视觉检测中人体所有肢体的位置与朝向），因此直接预测整个状态向量 $p(x | f(D))$ 很难。我们可以改为训练只预测**部分状态**的判别式检测器 $p(x_k | f_k(D))$ ，比如只预测人脸位置。于是可使用如下提议：

$$q(x' | x, D) = \pi_0 q_0(x' | x) + \sum_k \pi_k q_k(x'_k | f_k(D)) \quad (24.56)$$

其中 q_0 是标准的与数据无关的提议（如随机游走），而 q_k 只更新状态空间的第 k 个分量。为了进一步提效，判别式提议最好能**对多个变量做联合改变**，但这往往很难。

整体过程是一种“**生成并检验**”：判别式提议 $q(x' | x)$ 生成新假设，然后通过计算后验比 $\frac{p(x'|D)}{p(x|D)}$ 来“检验”新假设是否更好。加入退火步骤后，可以把算法改成寻找后验模态；这称为**模拟退火**，见 24.6.1 节。使用该寻模版本的一个好处是无需保证提议分布**可逆**。

24.3.4 自适应 MCMC

可以在算法运行过程中**改变提议的参数**以提升效率，这称为**自适应 MCMC**。这允许我们一开始用较宽的协方差（便于在空间中大步移动直到找到模态），随后逐步**收窄协方差**来在模态附近进行细致探索。

不过必须小心不要破坏**马尔可夫性**；因此提议的参数不应依赖于链的**全部历史**。事实证明，一个充分条件是让**自适应逐渐“消退”**（faded out），即随时间慢慢减弱。细节可见 (Andrieu and Thoms, 2008)。

24.3.5 初始化与跨模态跳转

必须从一个**具有非零概率**的初始状态开始 MCMC。若模型有确定性约束，找到这样一个“合法配置”本身就可能是个难题。因此常见做法是用优化器找到一个**局部模态**来初始化 MCMC。

在某些领域（尤其离散状态空间），相较于围绕一个局部模态附近反复探索，把计算时间用在**多次重启优化器**并对这些模态做平均更有效。然而，在**连续**状态空间中，单个模态所占体积可以忽略不计（见 5.2.1.3 节），因此需要在每个模态附近进行**局部探索**，以便覆盖足够的后验概率质量。

24.3.6 为什么 MH 有效

要证明 MH 过程会从 p^* 产生样本，我们需要用到一些马尔可夫链理论，因此务必先读

17.2.3 节。

MH 算法定义了如下转移矩阵（核）：

$$p(x' | x) = \begin{cases} q(x' | x) r(x' | x), & x' \neq x \\ q(x | x) + \sum_{y \neq x} q(y | x)(1 - r(y | x)), & x' = x \end{cases} \quad (24.57)$$

这来自分类讨论：从 x 移到 x' 必须是先提出 x' （概率 $q(x' | x)$ ），且被接受（概率 $r(x' | x)$ ）；否则就留在 x ，要么因为提议的就是 x （概率 $q(x | x)$ ），要么因为提议了别的 y （概率 $q(y | x)$ ）但被拒绝（概率 $1 - r(y | x)$ ）。

我们来分析这条马尔可夫链。回忆 17.2.3.4 节：若一条链满足**详细平衡**（detailed balance）

$$p(x' | x) p^*(x) = p(x | x') p^*(x') \quad (24.58)$$

则 p^* 是其平稳分布。我们的目标是证明：MH 定义的转移函数满足详细平衡，从而 p^* 是其平稳分布。（如果式 24.58 成立，我们就说 p^* 是相对于该马尔可夫转移核 q 的**不变分布**。）

定理 24.3.1。 若由 MH 算法（式 24.57）定义的转移矩阵是**遍历且不可约的**，则 p^* 是其唯一的极限分布。

证明。 取两状态 x 与 x' 。要么

$$p^*(x) q(x' | x) < p^*(x') q(x | x') \quad (24.59)$$

要么

$$p^*(x) q(x' | x) > p^*(x') q(x | x') \quad (24.60)$$

（对连续分布，平局发生概率为 0，可忽略。）不失一般性，设 $p^*(x) q(x' | x) > p^*(x') q(x | x')$ 。于是

$$\alpha(x' | x) = \frac{p^*(x') q(x | x')}{p^*(x) q(x' | x)} < 1 \quad (24.61)$$

因此 $r(x' | x) = \alpha(x' | x)$ ，而 $r(x | x') = 1$ 。

现在，从 x 到 x' 必须先提议 x' 再被接受，所以

$$p(x' | x) = q(x' | x) r(x' | x) = q(x' | x) \frac{p^*(x') q(x | x')}{p^*(x) q(x' | x)} = \frac{p^*(x')}{p^*(x)} q(x | x') \quad (24.62)$$

于是

$$p^{(*)}(x)p(x' | x) = p^{(*)}(x')q(x | x') \quad (24.63)$$

反向概率为

$$p(x | x') = q(x | x')r(x | x') = q(x | x') \quad (24.64)$$

因为 $r(x | x') = 1$ 。将其代入式 (24.63) 得

$$p^{(*)}(x)p(x' | x) = p^{(*)}(x')p(x | x') \quad (24.65)$$

故相对于 $p^{(*)}$ 成立详细平衡。于是由定理 17.2.3, $p^{(*)}$ 为平稳分布；又由定理 17.2.2, 因该链遍历且不可约，该分布唯一。

24.3.7 可逆跳（跨维）MCMC

设我们有一组参数数量不同的模型，例如混合模型的混合分量数未知。记模型为 m , 其未知量（如参数）记为 $x_m \in X_m$ （如 $X_m = \mathbb{R}^{n_m}$, 其中 n_m 是模型 m 的维度）。在不同维度的空间中采样称为跨维 MCMC (Green, 2003)。我们可以采样模型指示 $m \in \{1, \dots, M\}$, 并从笛卡尔积空间 $\prod_{m=1}^M X_m$ 采样所有参数，但这非常低效。更节省的是在并集空间 $X = \bigcup_{m=1}^M \{m\} \times X_m$ 中采样，此时只关心当前激活模型的参数。

困难出现在在不同维度的模型之间移动时。麻烦在于计算 MH 接受比时，我们在比较定义在不同维度空间的密度，这没有意义——就像拿一个球体和一个圆去比。Green (1998) 提出的解决方案称为可逆跳 MCMC (RJMCMC)：通过用额外随机变量扩充低维空间，使两个空间拥有共同的测度。

遗憾的是我们没有篇幅给出细节。简单地说，该方法在理论上可行，但实践起来有点棘手。然而，如果连续参数可以被积分消去（得到所谓折叠 RJMCMC），许多难点就消失了，因为我们只剩下离散状态空间，不必担心测度变换。例如，Denison 等 (2002) 包含了很多将折叠 RJMCMC 应用于自适应基函数模型贝叶斯推断的例子：他们从一个固定候选集合（如以数据点为中心）中采样基函数，并把其他参数解析积分掉。这提供了使用 RVM 或 SVM 的一种贝叶斯替代。

通俗解释（一步步讲清）

1) MH 在干嘛？一句话版

它让你“先大胆提议，再审慎接受”。先按某个提议分布 $q(x' | x)$ 试着跳到新位置，再按接受概率决定“收不收”这次跳。这样长期来看，到各处的“逗留时间”就按目标分布 $p^{(*)}(x)$ 来分配。

2) 为什么比吉布斯更通用？

吉布斯要能写出每个变量的完整条件分布并从中抽样，有时做不到（如逻辑回归的参数后

验没有好看的条件分布）。MH 只需要能算未归一化的后验 $\tilde{p}(x)$ 的数值（哪怕不知道归一化常数 Z ），就能干活。

3) 接受概率的直觉

- **对称提议**: $r = \min(1, \frac{p^*(x')}{p^*(x)})$ 。**更高就一定去；更低也偶尔去**（避免卡在局部峰）。
- **非对称提议**: 再乘个Hastings 校正，补偿“提议本身偏向”的影响。

4) 为什么不追求 100% 接受率？

全都接受通常意味着“几乎不动”（比如方差太小），探索性差。经验上把接受率调到 25%–40%（尤其在高斯型目标时）更能**平衡“动得远”与“别太常被拒”**。

5) 怎么选/调提议？

- **随机游走高斯**: $q = N(x' | x, \Sigma)$ 。 Σ 设太小→只在原地打转；太大→提议常被拒。
- **用 Hessian 设形状**: 在局部模态 \hat{x} 处算 H ，提议协方差设为 H^{-1} 的尺度版，方向/尺度都更合适；理论上**最优尺度** $s^2 \approx 2.38^2/D$ ，验收率约 0.234。
- **独立提议/混合提议**: 可用独立于当前点的 $q(x')$ ，或把好几种 q_k 混起来用。
- **数据驱动提议**: 让 q 看数据 D ，先用判别模型“猜”局部（如猜人脸位置），再用 MH 的接受步骤把关。
- **自适应 MCMC**: 边跑边调 Σ 等参数（但要**逐渐减弱**适应，别让链依赖整个历史）。

6) 初始化和多模态

- 初始状态必须是**合法的**（概率不为 0）。
- 连续空间里，一个模态本身体积几乎 0，所以不仅要找到模态，还得在附近**好好探索**。
- 离散空间时，**多次重启+平均**有时更划算。

7) “MH 为何有效”的关键理由

它构造的转移满足**详细平衡**:

$$p(x' | x)p^*(x) = p(x | x')p^*(x')$$

因此 p^* 是平稳分布；若链**不可约且遍历**，就会收敛到它（且唯一）。这就是“为何走久了样本就像来自 p^* ”的数学保证。

8) 跨维度 (RJMCMC)

当模型维度会变（比如混合簇数不定）时，不能直接比不同维空间的密度。RJMCMC 的做法是**给低维那边补几个随机量**，把两边放到“同一可比空间”，再做 MH。若能把连续参数**积分掉**（折叠法），问题大为简化。

要点总结

- **MH 核心**: 先按提议 $q(x' | x)$ 生成候选，再按接受概率（含 Hastings 校正）决定去不去，长期样本分布就是 p^* 。
- **不需要归一化常数**: 只要能算未归一化密度 $\tilde{p}(x)$ ，就能跑。
- **吉布斯是 MH 特例**: 用完整条件做提议 \Rightarrow 接受率 100%，但不一定混合快。
- **调参数重于高接受**: 目标通常把接受率调到 25%–40%；高斯随机游走在高维的**最优尺度近似** $s^2 = 2.38^2/D$ ，验收率约 0.234。
- **更强提议**: 利用 Hessian 的形状、做混合提议、做数据驱动提议、或自适应（逐渐减弱）以提速。
- **初始化与多模态**: 要从合法点启动；连续空间需在模态附近充分探索；离散空间时多次重启可能更值。
- **数学保证**: MH 满足详细平衡 $\Rightarrow p^*$ 为平稳分布；在**不可约、遍历**条件下收敛到它。
- **跨维模型**: 用 RJMCMC 处理维度变化；若能折叠（积分掉连续参数）更容易。

如果你愿意，我可以用一个**小数值例子**（比如从“两峰高斯混合”采样）演示不同方差的随机游走提议对接受率与探索的影响，并画出轨迹对比图，帮助直观体会怎么“调到刚刚好”。

这一节主要在讲什么？我不太懂

已思考 12s

没事～这节 (24.3) 核心就三句话：**先提议、再验收、长期就对**。下面用大白话把主线拎清👉

一句话概括

Metropolis–Hastings (MH) 是一种通用的 MCMC：从当前点按某个**提议分布**先提一个候选点，再用**接受概率**决定要不要跳过去。只要这样反复做，样本长期分布就会等于目标分布 $p^*(x)$ 。

它为啥重要 (解决啥问题)

- **比吉布斯更通用**: 很多模型（如逻辑回归）不容易直接写出“每个变量的条件分布”来抽样；MH 不需要这些，只要能算未归一化的后验 $\tilde{p}(x)$ （不用知道归一化常数 Z ！）就能跑。
- **能下坡**: 不像贪心只往更可能的地方去，MH 允许偶尔“下坡”，避免卡在局部峰。

算法怎么走 (极简版)

1. 提议：从 $q(x' | x)$ 抽一个候选 x' (常用：以当前点为中心的高斯“随机游走”)。
2. 验收：算
 - 对称提议： $r = \min\{1, p^{(*)}(x')/p^{(*)}(x)\}$
 - 非对称： $r = \min\{1, [p^{(*)}(x')q(x | x')]/[p^{(*)}(x)q(x' | x)]\}$ (Hastings 校正)
然后以概率 r 接受；否则留在原地。
3. 重复很多轮 (前面有烧入期要丢)。

和吉布斯的关系

吉布斯 = MH 的特例：提议就用“完整条件分布” $p(x_i | x_{-i})$ ，这时接受率=100%。但吉布斯一次只动一个坐标，相关性强时混合慢；MH 可以设计更大胆的联合提议。

提议怎么选 & 怎么调

- 随机游走高斯：方差太小→几乎不动；太大→提议常被拒。
实务上把接受率调到 25%–40% 往往更好 (探索 vs. 通过 的折中)。
- 用曲率 (Hessian) 定形状：让提议协方差 $\propto H^{-1}$ ，各方向步子更合适。高维下有经验尺度 $s^2 \approx 2.38^2/D$ (接受率≈0.234)。
- 混合提议：把几种 q_k 按权重混起来用。
- 数据驱动提议：让 $q(x' | x, D)$ 也看数据特征，先“猜好位置”，再用 MH 把关。
- 自适应 MCMC：边跑边调 q 的参数，但要逐渐减弱适应，别依赖整段历史，避免破坏马尔可夫性。

初始化与多模态

- 得从合法的初始点起步 (概率不为 0)。
- 连续空间里，一个模态体积很小，只找到峰还不够，要在附近充分探索。
- 离散空间时，多次重启+对结果平均有时更划算。

为什么 MH 有效 (可信的数学理由)

它把转移概率设计成满足详细平衡： $p(x' | x)p^{(*)}(x) = p(x | x')p^{(*)}(x')$ 。若链还不可约且遍历，就会收敛到唯一的平稳分布 $p^{(*)}$ ——所以“走久了就对了”。

维度会变怎么办 (RJMCMC)

模型大小可变 (如簇数未知) 时，用可逆跳 MCMC 把不同维度的比较放到同一“可比空间”，必要时把连续参数积分掉 (折叠)，问题更简单。

这节要带走的点

- MH = 先提议、再验收；能用未归一化后验，超通用。
- 吉布斯是特例（接受率 $100\% \neq$ 混合快）。
- 调好提议比追高接受率更重要：常见目标接受率 25%–40%，高维随机游走用 $2.38^2/D$ 尺度是个好起点。
- 可用 Hessian/混合/数据驱动/自适应等招提升效率。
- 详细平衡 + 不可约遍历 \Rightarrow 收敛到目标分布。
- RJMCMC 处理模型维度变化，折叠可进一步简化。

要不要我用一个“双峰一维高斯混合”的小例子，演示不同方差下的轨迹与接受率差异？看一张图就更直观～

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24.4 Speed and accuracy of MCMC

In this section, we discuss a number of important theoretical and practical issues to do with MCMC.

24.4.1 The burn-in phase

We start MCMC from an arbitrary initial state. As we explained in Section 17.2.3, only when the chain has “forgotten” where it started from will the samples be coming from the chain’s stationary distribution. Samples collected before the chain has reached its stationary distribution do not come from p_* , and are usually thrown away. The initial period, whose samples will be ignored, is called the burn-in phase.

For example, consider a uniform distribution on the integers $\{0, 1, \dots, 20\}$. Suppose we sample from this using a symmetric random walk. In Figure 24.9, we show two runs of the algorithm. On the left, we start in state 10; on the right, we start in state 17. Even in this small problem it takes over 100 steps until the chain has “forgotten” where it started from.

It is difficult to diagnose when the chain has burned in, an issue we discuss in more detail below. (This is one of the fundamental weaknesses of MCMC.) As an interesting example of what can happen if you start collecting

samples too early, consider the Potts model. Figure 24.10(a), shows a sample after 500 iterations of Gibbs sampling. This suggests that the model likes medium-sized regions where the label is the same, implying the model would make a good prior for image segmentation. Indeed, this was suggested in the original Gibbs sampling paper (Geman and Geman 1984).

However, it turns out that if you run the chain long enough, you get isolated speckles, as in Figure 24.10(b). The results depend on the coupling strength, but in general, it is very hard to find a setting which produces nice medium-sized blobs: most parameters result in a few super-clusters, or lots of small fragments. In fact, there is a rapid phase transition between these two regimes. This led to a paper called "The Ising/Potts model is not well suited to segmentation tasks" (Morris et al. 1996). It is possible to create priors more suited to image segmentation (e.g., (Sudderth and Jordan 2008)), but the main point here is that sampling before reaching convergence can lead to erroneous conclusions.

24.4.2 Mixing rates of Markov chains

The amount of time it takes for a Markov chain to converge to the stationary distribution, and forget its initial state, is called the mixing time. More formally, we say that the mixing time from state x_0 is the minimal time such that, for any constant $\epsilon > 0$, we have that

$$\tau_{x_0} = \min\{t : \|\delta_{x_0}(x)T_t - p^*\|_1 \leq \epsilon\} \quad (24.66)$$

where $\delta_{x_0}(x)$ is a distribution with all its mass in state x_0 , T is the transition matrix of the chain (which depends on the target p^* and the proposal q), and $\delta_{x_0}(x)T_t$ is the distribution after t steps. The mixing time of the chain is defined as

$$\tau = \max_{x_0} \tau_{x_0} \quad (24.67)$$

The mixing time is determined by the eigengap $\gamma = \lambda_1 - \lambda_2$, which is the difference of the first and second eigenvalues of the transition matrix. In particular, one can show that

$$\tau \leq O(1/\gamma \log n) \quad (24.68)$$

where n is the number of states. Since computing the transition matrix can be hard to do, especially for high dimensional and/or continuous state spaces, it is useful to find other ways to estimate the mixing time.

An alternative approach is to examine the geometry of the

state space. For example, consider the chain in Figure 24.11. We see that the state space consists of two “islands”, each of which is connected via a narrow “bottleneck”. (If they were completely disconnected, the chain would not be ergodic, and there would no longer be a unique stationary distribution.) We define the conductance φ of a chain as the minimum probability, over all subsets of states, of transitioning from that set to its complement:

$$\varphi = \min_{S: 0 \leq p^*(S) \leq 0.5} \sum_{x \in S, x' \in S^c} T(x \rightarrow x') / p^*(S), \quad (24.69)$$

One can show that

$$\tau \leq O(1/\varphi^2 \log n) \quad (24.70)$$

Hence chains with low conductance have high mixing time. For example, distributions with well-separated modes usually have high mixing time. Simple MCMC methods often do not work well in such cases, and more advanced algorithms, such as parallel tempering, are necessary (see e.g., (Liu 2001)).

24.4.3 Practical convergence diagnostics

Computing the mixing time of a chain is in general quite difficult, since the transition matrix is usually very hard to compute. In practice various heuristics have been proposed to diagnose convergence — see (Geyer 1992; Cowles and Carlin 1996; Brooks and Roberts 1998) for a review. Strictly speaking, these methods do not diagnose convergence, but rather non-convergence. That is, the method may claim the chain has converged when in fact it has not. This is a flaw common to all convergence diagnostics, since diagnosing convergence is computationally intractable in general (Bhatnagar et al. 2010).

One of the simplest approaches to assessing when the method has converged is to run multiple chains from very different overdispersed starting points, and to plot the samples of some variables of interest. This is called a trace plot. If the chain has mixed, it should have “forgotten” where it started from, so the trace plots should converge to the same distribution, and thus overlap with each other.

Figure 24.12 gives an example. We show the traceplot for x which was sampled from a mixture of two 1D Gaussians using four different methods: MH with a symmetric

Gaussian proposal of variance $\sigma^2 \in \{1, 8, 500\}$, and Gibbs sampling. We see that $\sigma^2 = 1$ has not mixed, which is also evident from Figure 24.7(a), which shows that a single chain never leaves the area where it started. The results for the other methods indicate that the chains rapidly converge to the stationary distribution, no matter where they started. (The sticky nature of the $\sigma^2 = 500$ proposal is very evident. This reduces the computational efficiency, as we discuss below, but not the statistical validity.)

24.4.3.1 Estimated potential scale reduction (EPSR)

We can assess convergence more quantitatively as follows. The basic idea is to compare the variance of a quantity within each chain to its variance across chains. More precisely, suppose we collect S samples (after burn-in) from each of C chains of D variables, x_{isc} , $i=1:D$, $s=1:S$, $c=1:C$. Let ysc be a scalar quantity of interest derived from $x_{1:D,s,c}$ (e.g., $ysc = x_{isc}$ for some chosen i). Define the within-sequence mean and overall mean as

$$\bar{y}_c = \frac{1}{S} \sum_{s=1}^S y_{sc}, \quad \bar{y}_{..} = \frac{1}{C} \sum_{c=1}^C \bar{y}_c \quad (24.71)$$

Define the between-sequence and within-sequence variance as

$$B = \frac{1}{C} \sum_{c=1}^C (\bar{y}_c - \bar{y}_{..})^2, \quad W = \frac{1}{S} \sum_{s=1}^S (\bar{y}_{sc} - \bar{y}_c)^2 \quad (24.72)$$

We can now construct two estimates of the variance of y . The first estimate is W : this should underestimate $\text{var}[y]$ if the chains have not ranged over the full posterior. The second estimate is

$$\hat{V} = S - 1 S W + 1 S B \quad (24.73)$$

This is an estimate of $\text{var}[y]$ that is unbiased under stationarity, but is an overestimate if the starting points were overdispersed (Gelman and Rubin 1992). From this, we can define the following convergence diagnostic statistic, known as the estimated potential scale reduction or EPSR:

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}} \quad (24.74)$$

This quantity, which was first proposed in (Gelman and Rubin 1992), measures the degree to which the posterior variance would decrease if we were to continue sampling in the $S \rightarrow \infty$ limit. If $\hat{R} \approx 1$ for any given quantity, then that estimate is reliable (or at least is not unreliable). The \hat{R} values for the four samplers in Figure 24.12 are 1.493, 1.039, 1.005 and 1.007. So this diagnostic has correctly identified that the sampler using the first ($\sigma^2 = 1$) proposal

is untrustworthy.

24.4.4 Accuracy of MCMC

The samples produced by MCMC are auto-correlated, and this reduces their information content relative to independent or “perfect” samples. We can quantify this as follows. Suppose we want to estimate the mean of $f(X)$, for some function f , where $X \sim p()$. Denote the true mean by $f_* = E[f(X)]$ (24.75)

A Monte Carlo estimate is given by

$$f = \frac{1}{S} \sum_{s=1}^S f_s \quad (24.76)$$

where $f_s = f(x_s)$ and $x_s \sim p(x)$. An MCMC estimate of the variance of this estimate is given by

$$\text{VarMCMC}[f] = E[(f - f_*)^2] \quad (24.77)$$

$$= E[\frac{1}{S} \sum_{s=1}^S (f_s - f_*)^2] \quad (24.78)$$

$$= \frac{1}{S^2} \sum_{s=1}^S (f_s - f_*)^2 + \frac{1}{S^2} \sum_{s=1}^S (f_s - f_*)(f_t - f_*) \quad (24.79)$$

$$= \text{VarMC}(f) + \frac{1}{S^2} \sum_{s=1}^S (f_s - f_*)(f_t - f_*) \quad (24.80)$$

where the first term is the Monte Carlo estimate of the variance if the samples weren’t correlated, and the second term depends on the correlation of the samples. We can measure this as follows. Define the sample-based auto-correlation at lag t of a set of samples f_1, \dots, f_S as follows:

$$\rho_t = \frac{\sum_{s=1}^{S-t} (f_s - f)(f_{s+t} - f)}{\sum_{s=1}^S (f_s - f)^2} \quad (24.81)$$

This is called the autocorrelation function (ACF). This is plotted in Figure 24.13 for our four samplers for the Gaussian mixture model. We see that the ACF of the Gibbs sampler (bottom right) dies off to 0 much more rapidly than the MH samplers, indicating that each Gibbs sample is “worth” more than each MH sample.

A simple method to reduce the autocorrelation is to use thinning, in which we keep every n ’th sample. This does not increase the efficiency of the underlying sampler, but it does save space, since it avoids storing highly correlated samples.

We can estimate the information content of a set of samples by computing the effective sample size (ESS) S_{eff} , defined by

$$S_{\text{eff}} = \frac{\text{VarMC}(f)}{\text{VarMCMC}(f)} \quad (24.82)$$

From Figure 24.12, it is clear that the effective sample size of the Gibbs sampler is higher than that of the other samplers (in this example).

24.4.5 How many chains?

A natural question to ask is: how many chains should we run? We could either run one long chain to ensure convergence, and then collect samples spaced far apart, or we could run many short chains, but that wastes the burnin time. In practice it is common to run a medium number of chains (say 3) of medium length (say 100,000 steps), and to take samples from each after discarding the first half of the samples. If we initialize at a local mode, we may be able to use all the samples, and not wait for burn-in.

已思考 11s

下面我先给出逐句完整译文（不遗漏），再用通俗解释把关键概念讲清楚，最后给出要点总结方便复习。

逐句完整译文（不遗漏）

24.4 MCMC 的速度与准确性

在本节中，我们讨论若干与 MCMC 有关的重要理论与实践问题。

24.4.1 烧入阶段 (burn-in phase)

我们从一个任意的初始状态开始 MCMC。正如我们在 17.2.3 节解释的那样，只有当链已经“忘记”它从哪里开始之后，样本才会来自该链的平稳分布。在链达到其平稳分布之前收集的样本并不来自 p^* ，通常会被丢弃。这个初始时期（其样本将被忽略）称为**烧入阶段 (burn-in)**。

例如，考虑整数 $\{0, 1, \dots, 20\}$ 上的均匀分布。假设我们用一个对称的随机游走从这里采样。在图 24.9 中，我们展示该算法的两次运行。左图从状态 10 开始；右图从状态 17 开始。即便在这个很小的问题中，也需要超过 100 步，链才能“忘记”其起点。

诊断链何时完成了烧入是困难的，我们将在下文更详细地讨论这个问题。（这是 MCMC 的一个根本弱点。）一个有趣的例子表明，如果你过早开始收集样本，会发生什么：考虑 Potts 模型。图 24.10(a) 显示了吉布斯采样 500 次迭代后的一个样本。这暗示该模型喜欢中等大小、标签相同的区域，这意味着该模型会成为图像分割的一个良好先验。事实上，这在最初的吉布斯采样论文 (Geman and Geman 1984) 中就被提出了。

然而，事实证明，如果你运行该链足够长，你会得到孤立的斑点，如图 24.10(b)。结果依赖于耦合强度，但一般来说，很难找到能产生漂亮中等大小斑块的设定：大多数参数会导致少数超大簇，或许多小碎片。实际上，这两种状态之间存在一个快速的相变。这导致了一篇论文，题为“Ising/Potts 模型并不适合分割任务” (Morris et al. 1996)。可以构造更适

合图像分割的先验（例如，Sudderth and Jordan 2008），但这里的要点是：**在收敛前进行采样会得出错误的结论。**

24.4.2 马尔可夫链的混合速率

马尔可夫链收敛到平稳分布并忘记其初始状态所需的时间称为**混合时间**（mixing time）。更形式地，我们说，从状态 x_0 的混合时间是使得对于任意常数 $\varepsilon > 0$ ，满足

$$\tau_\varepsilon(x_0) = \min \{t : \| \delta_{x_0}(x) T^t - p^{*} \|_1 \leq \varepsilon\} \quad (24.66)$$

的最长时间；其中 $\delta_{x_0}(x)$ 是把全部质量放在状态 x_0 的分布， T 是该链的转移矩阵（取决于目标 p^{*} 与提议 q ），而 $\delta_{x_0}(x)T^t$ 是 t 步之后的分布。链的混合时间定义为

$$\tau = \max_{x_0} \tau_\varepsilon(x_0) \quad (24.67)$$

混合时间由**特征值间隙** $\gamma = \lambda_1 - \lambda_2$ 决定，即转移矩阵第一与第二特征值之差。特别地，可以证明

$$\tau \leq O\left(\frac{1}{\gamma} \log n\right) \quad (24.68)$$

其中 n 是状态数。由于计算转移矩阵可能很难（尤其对高维和/或连续状态空间），因此找到其他方式估计混合时间是有用的。

另一种方法是检查状态空间的几何结构。例如，考虑图 24.11 中的链。我们看到状态空间由两个“岛”组成，它们通过一个狭窄的“瓶颈”相连。（如果它们完全不相连，该链将不是遍历的，也就不再有唯一的平稳分布。）我们把链的**导通率**（conductance） ϕ 定义为在所有状态子集上，从该集合跳到其补集的最小概率：

$$\phi = \min_{S: 0 \leq p^{*}(S) \leq 0.5} \frac{\sum_{x \in S, x' \in S^c} T(x \rightarrow x')}{p^{*}(S)} \quad (24.69)$$

可以证明

$$\tau \leq O\left(\frac{1}{\phi^2} \log n\right) \quad (24.70)$$

因此，**导通率低的链具有很高的混合时间**。例如，**模态分离良好的分布通常有很高的混合时间**。简单的 MCMC 方法在此类情形下往往效果不好，需要更高级的算法，如**并行退火**（parallel tempering）（参见如 Liu 2001）。

24.4.3 实用的收敛诊断

一般来说，计算一条链的混合时间相当困难，因为转移矩阵通常极其难以计算。实践中提出了各种启发式方法来诊断收敛——综述可见 Geyer (1992)、Cowles and Carlin (1996)、

Brooks and Roberts (1998)。严格地说，这些方法并非诊断“已收敛”，而是诊断“未收敛”。也就是说，方法可能会宣称链已收敛，而实际上它并未收敛。此缺陷是所有收敛诊断的共性，因为一般而言诊断收敛在计算上是不可判定/不可处理的 (Bhatnagar et al. 2010)。

评估方法何时收敛的最简单途径之一是：从**非常不同且分散 (overdispersed)** 的**初始点启动多条链，并绘制某些感兴趣变量的样本图。这称为轨迹图 (trace plot)**。如果链已混合，它应当已经“忘记起点”，因此这些轨迹图应收敛到同一分布，并彼此重叠。

图 24.12 给出了一个例子。我们展示了对 x 的轨迹图，它来自用四种不同方法从两个一维高斯的混合分布采样：使用方差 $\sigma^2 \in \{1, 8, 500\}$ 的对称高斯提议的 MH，以及吉布斯采样。我们看到 $\sigma^2 = 1$ 并未混合，这在图 24.7(a) 中也很明显：单条链从未离开其起始区。其他方法的结果表明，不论从何处开始，这些链都能快速收敛到平稳分布。 $(\sigma^2 = 500)$ 的“粘滞性”非常明显。这会降低**计算效率** (如下所述)，但不会影响**统计有效性**。)

24.4.3.1 估计的潜在尺度缩减 (EPSR)

我们可以如下更定量地评估收敛。基本思想是比较某个量在**链内的方差与链间的方差**。更精确地，假设我们从 C 条链的 D 个变量中，在烧入后各收集 S 个样本，记作 x_{isc} , $i = 1:D$, $s = 1:S$, $c = 1:C$ 。令 y_{sc} 为从 $x_{1:D,s,c}$ 派生的某个标量感兴趣量 (例如对某个选定的 i ，取 $y_{sc} = x_{isc}$)。定义链内均值与总体均值为

$$\bar{y}_{\cdot c} = \frac{1}{S} \sum_{s=1}^S y_{sc}, \quad \bar{y}_{..} = \frac{1}{C} \sum_{c=1}^C \bar{y}_{\cdot c} \quad (24.71)$$

定义链间方差与链内方差为

$$B = \frac{S}{C-1} \sum_{c=1}^C (\bar{y}_{\cdot c} - \bar{y}_{..})^2, \quad W = \frac{1}{C} \sum_{c=1}^C \frac{1}{S-1} \sum_{s=1}^S (y_{sc} - \bar{y}_{\cdot c})^2 \quad (24.72)$$

现在我们可以构造两个对 y 方差的估计。第一种估计是 W ：如果链尚未覆盖整个后验，它倾向于**低估** $\text{var}[y]$ 。第二种估计是

$$\hat{V} = \frac{S-1}{S} W + \frac{1}{S} B \quad (24.73)$$

这是在平稳条件下对 $\text{var}[y]$ 的**无偏估计**，但如果起点**过分分散**，它会**高估** (Gelman and Rubin 1992)。据此，我们可以定义如下收敛诊断统计量，称为**估计的潜在尺度缩减**或 EPSR：

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}} \quad (24.74)$$

这个量最早由 Gelman and Rubin (1992) 提出，用来度量：如果我们继续采样至 $S \rightarrow \infty$ ，后验方差还会缩减多少。若对某个量 $\hat{R} \approx 1$ ，则该估计是可靠的（或至少不是不可靠的）。图 24.12 中四个采样器的 \hat{R} 值分别为 1.493、1.039、1.005、1.007。因此该诊断正确识别出使用第一个 ($\sigma^2 = 1$) 提议的采样器不值得信任。

24.4.4 MCMC 的准确性

MCMC 产生的样本是**自相关的**，这会相对独立（或“完美”）样本降低其信息含量。我们可以如下量化这一点。假设我们想估计某个函数 f 在 $X \sim p(\cdot)$ 下的均值。把真实均值记为

$$f^* = E[f(X)] \quad (24.75)$$

一个蒙特卡洛估计为

$$\bar{f} = \frac{1}{S} \sum_{s=1}^S f_s \quad (24.76)$$

其中 $f_s = f(x_s)$ ，且 $x_s \sim p(x)$ 。该估计的一个 MCMC 方差估计为

$$\text{Var}_{\text{MCMC}}[\bar{f}] = E[(\bar{f} - f^*)^2] \quad (24.77)$$

$$= E \left[\left(\frac{1}{S} \sum_{s=1}^S (f_s - f^*) \right)^2 \right] \quad (24.78)$$

$$= \frac{1}{S^2} E \left[\sum_{s=1}^S (f_s - f^*)^2 + \sum_{s \neq t} (f_s - f^*)(f_t - f^*) \right] \quad (24.79)$$

$$= \text{Var}_{\text{MC}}(\bar{f}) + \frac{1}{S^2} \sum_{s \neq t} E[(f_s - f^*)(f_t - f^*)] \quad (24.80)$$

其中第一项是若样本**不相关**时的蒙特卡洛方差估计，而第二项取决于样本的相关性。我们可以如下度量它。定义样本集合 f_1, \dots, f_S 在滞后 t 处的基于样本的**自相关**为

$$\rho_t = \frac{\frac{1}{S-t} \sum_{s=1}^{S-t} (f_s - \bar{f})(f_{s+t} - \bar{f})}{\frac{1}{S-1} \sum_{s=1}^S (f_s - \bar{f})^2} \quad (24.81)$$

这称为**自相关函数 (ACF)**。图 24.13 为我们在高斯混合模型上的四个采样器的 ACF。可以看到，吉布斯采样器（右下）的 ACF 更快衰减到 0，提示**每个吉布斯样本“更有用”**（信息含量高于 MH 样本）。

一种简单的减小自相关的方法是**抽稀 (thinning)**，即只保留每隔 n 个的样本。这并不会提升底层采样器的效率，但能节省存储，因为避免保存高度相关的样本。

我们可以通过计算**有效样本量 (ESS)** S_{eff} 来估计样本的信息含量，定义为

$$S_{\text{eff}} = \frac{\text{Var}_{\text{MC}}(\bar{f})}{\text{Var}_{\text{MCMC}}(f)} \quad (24.82)$$

从图 24.12 可以看出（在该例中）**吉布斯采样器的有效样本量更高**，超过其他采样器。

24.4.5 应该跑多少条链？

一个自然的问题是：我们应该跑多少条链？我们可以跑一条很长的链以确保收敛，然后隔很久取一个样本；也可以跑许多短链，但那会浪费烧入时间。实践中，通常做法是跑一个**中等数量**（比如 3 条）的链，每条**中等长度**（比如 100,000 步），并在丢弃前一半样本后，从每条链取样。如果我们用一个**局部模态**来初始化，则可能可以使用**全部样本**，而不必等待烧入。

通俗解释（一步步讲清）

1) 为什么要“烧入”？

刚开始的样本带着“初始点的味道”，**还不像目标分布**。烧入就是“热身期”：把这段样本丢掉，让链先把起点“忘了”。如果太早用样本下结论，就可能像 Potts 模型的例子一样，**被误导**（以为会产生中等大小的块，结果跑久了发现不是）。

2) 什么叫“混合”（mixing）？怎么快/慢？

- **混合时间**：大概就是“多久后样本才像目标分布”。
- 受两件事影响：
 - **特征值间隙** γ ：越大，混合越快。
 - **导通率** ϕ ：状态空间像两座岛，中间一个小桥（瓶颈）， ϕ 很小 \Rightarrow 来回穿越难 \Rightarrow 混合慢。
- 对“多峰且相距远”的分布，简单 MCMC 往往**混不开**，要用像**并行退火**这样的招数。

3) 我怎么知道“差不多收敛了”？（实操诊断）

- **画轨迹图**：从不同初值启动多条链。如果“线条堆到一块”、看起来都在同一分布里抖动，说明混得差不多了。
- **EPSR (\hat{R})**：比较“链内方差” vs “链间方差”。如果 $\hat{R} \approx 1$ ，通常更可信；比如前文例子里 $\sigma^2 = 1$ 的提议给出 $\hat{R} = 1.493$ ，明显不靠谱。

小提醒：这些方法**只能证明“不好”（未收敛）**，很难严格证明“已经好”（已收敛）。所以要多管齐下、保持谨慎。

4) MCMC 样本为什么“没那么值钱”？（准确性）

- 相邻样本相关 \Rightarrow 信息量打折。
- 看 ACF：衰得越快，相关越小，样本越“值钱”。例子里吉布斯的 ACF 衰得快，所以它的**有效样本量 (ESS) **更大。
- 抽稀（每隔 n 个取一个）能省存储，但**不会**让底层采样器更高效。

5) 到底跑几条链、每条多长？

- 一条超长链：省烧入次数，但要等。
 - 很多短链：烧入浪费多。
 - 经验：3 条左右、每条**中等长度**（如 10 万步），丢掉前一半；若从局部模态启，可能**不用烧入**。
-

要点总结

- **烧入 (burn-in)**：初期样本不可靠，要去；过早用样本可能误判模型行为（如 Potts 例子）。
- **混合时间**：由**特征值间隙 γ** 与**导通率 ϕ** 控制；瓶颈/多峰 \Rightarrow 混得慢，需要高级技巧（并行退火等）。
- **收敛诊断**：**多链轨迹图**与 EPSR ($\hat{R} \approx 1$ **更可信**) 是常用手段，但都可能“误报已收敛”，只能强力识别未收敛。
- **准确性与相关**：样本自相关会增大方差、降低信息量；看 ACF 与 ESS 评估“样本有多值钱”。抽稀仅省存储，不提效率。
- **链的数量与长度**：实践上常用“几条中等长度链 + 丢前半”，从模态初始化有时可省烧入。

注：文中 \hat{R} 的正规公式为 $\hat{R} = \sqrt{\hat{V}/W}$ ；如果你在别处看到排版异常（比如像“8”而不是根号），应以此式为准。

有需要的话，我可以给你做一张“多条链轨迹 + ACF + \hat{R} ”的小示例图，直观看 burn-in、混合与相关性怎么影响结果。

这一节主要在讲什么？我不太懂

已思考 7s

不慌～这节 (24.4) 的主线就是：**怎么判断 MCMC 够“快”、够“准”，以及该怎么做才靠谱**。五个点 

一句话概括

跑 MCMC 不是“跑就完了”，你得：先把**热身期 (burn-in)** 扔掉；看它混得快不快 (mixing)；用工具检查收敛；意识到样本**有相关、信息打折**；最后决定**跑几条链、多长**才合适。

这节在讲的5件事

1. 烧入 (burn-in) 要丢

刚开始的样本还“记得起点”，不代表目标分布，所以要丢掉一段热身期。太早用样本会得出错觉（文中 Potts 模型例子：500步看起来像会出“中号斑块”，但再跑久一点变成“零碎点”或“大团块”，完全两回事）。

2. 混合快慢 (mixing time)

混合时间 = 链需要多久忘记起点、像目标分布。

- 受**特征值间隙**（大→快）和**导通率**（瓶颈小桥→慢）影响。
- 多峰且相距远 ⇒ 混得慢；常需更高级技巧（如并行退火）而不是朴素的随机游走。

3. 收敛诊断（实操）

真正“证明已收敛”很难，所以我们用**能抓出“没收敛”**的招：

- **多链轨迹图 (trace plot)**：从截然不同的初值起跑，看曲线最后是否“堆在一起”。
- **EPSR (\hat{R})**：比链间方差和链内方差； \hat{R} 接近 1 更靠谱。例子里方差太小的提议给出 $\hat{R} \approx 1.49$ ，不可信。

4. 准确性=信息量（相关性会打折）

MCMC 相邻样本相关 ⇒ 单个样本没那么“值钱”。

- 看**自相关函数 ACF**：掉得越快越好。例子里吉布斯衰得快，每个样本更“值钱”。
- **抽稀 (thinning)**：每隔 n 个留一个，只是省存储，不提升采样器效率。
- **有效样本量 (ESS)**：把“相关样本”的信息量折算成“等价独立样本数”，ESS 越大越好。

5. 到底跑几条链？

- 一条超长链：烧入少，但等得久。
- 很多短链：烧入浪费多。
- 常用折中：**几条中等长度的链**（比如 3 条），各自**丢前半段后取样**。若能从**局部模态**初始化，有时几乎不用烧入。

带走的要点

- **先丢热身期**，别用太早的样本下结论。

- **混合速度受结构影响：**瓶颈/多峰 \Rightarrow 慢；必要时用更强的算法（如并行退火）。
- **用多链+轨迹图+EPSR做非收敛的预警；** $\hat{R} \approx 1$ 更放心。
- **样本相关 \rightarrow 信息打折：**看 ACF、算 ESS；抽稀只省空间，不提效率。
- **链数与长度用折中策略：**几条中等长度、各自丢前半；从模态起步可减少烧入需求。

如果你愿意，我可以给你做个小演示图（多链轨迹、ACF、 \hat{R} 、ESS）来直观看 burn-in、混合和相关性各自的影响。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24.5 Auxiliary variable MCMC

Sometimes we can dramatically improve the efficiency of sampling by introducing dummy auxiliary variables, in order to reduce correlation between the original variables. If the original variables are denoted by x , and the auxiliary variables by z , we require that $p(x, z) = p(x)$, and that $p(x, z)$ is easier to sample from than just $p(x)$. If we meet these two conditions, we can sample in the enlarged model, and then throw away the sampled z values, thereby recovering samples from $p(x)$. We give some examples below.

24.5.1 Auxiliary variable sampling for logistic regression

In Section 9.4.2, we discussed the latent variable interpretation of probit regression. Recall that this had the form

$$z_i = w^T x_i + i \quad (24.83)$$

$$i \sim N(0, 1) \quad (24.84)$$

$$y_i = 1 = I(z_i \geq 0) \quad (24.85)$$

We exploited this representation in Section 11.4.6, where we used EM to find an ML estimate. It is straightforward to convert this into an auxiliary variable Gibbs sampler (Exercise 24.6), since $p(w|D)$ is Gaussian and $p(z_i|x_i, y_i, w)$ is truncated Gaussian, both of which are easy to sample from.

Now let us discuss how to derive an auxiliary variable Gibbs sampler for logistic regression. Let i follow a logistic distribution, with pdf

$$p_{\text{Logistic}}() = e^{- (1 + e^{- })} \quad (24.86)$$

with mean $E [] = 0$ and variance $\text{var} [] = \pi^2/3$. The cdf

has the form $F(\beta) = \text{sigm}(\beta)$, which is the logistic function. Since $y_i = 1$ iff $w^T x_i + \beta > 0$, we have, by symmetry, that

$$\begin{aligned} p(y_i = 1 | x_i, w) &= \infty - w^T x_i f(\beta) d = w^T x_i - \infty f(-\beta) d = F(w^T x_i) \\ &= \text{sigm}(w^T x_i) \quad (24.87) \end{aligned}$$

as required.

We can derive an auxiliary variable Gibbs sampler by sampling from $p(z|w, D)$ and $p(w|z, D)$. Unfortunately, sampling directly from $p(w|z, D)$ is not possible. One approach is to define $\lambda_i \sim N(0, \lambda_i)$, where $\lambda_i = (2\psi_i)^2$ and $\psi_i \sim KS$, the Kolmogorov Smirnov distribution, and then to sample w, z, λ and ψ (Holmes and Held 2006).

A simpler approach is to approximate the logistic distribution by the Student distribution (Albert and Chib 1993). Specifically, we will make the approximation $\lambda_i \sim T(0, 1/v)$, where $v \approx 8$. We can now use the scale mixture of Gaussians representation of the Student to simplify inference. In particular, we write

$$\lambda_i \sim Ga(v/2, v/2) \quad (24.88)$$

$$\lambda_i \sim N(0, \lambda_i^{-1}) \quad (24.89)$$

$$z_i = w^T x_i + \lambda_i \quad (24.90)$$

$$y_i = 1 | z_i = I(z_i \geq 0) \quad (24.91)$$

All of the full conditionals now have a simple form; see Exercise 24.7 for the details.

Note that if we set $v = 1$, then $z_i \sim N(w^T x_i, 1)$, which is equivalent to probit regression (see Section 9.4). Rather than choosing between probit or logit regression, we can simply estimate the v parameter. There is no convenient conjugate prior, but we can consider a finite range of possible values and evaluate the posterior as follows:

$$p(v|\lambda) \propto p(v) \prod_{i=1}^n \Gamma(v/2)(v/2)^{v/2} \lambda_i^{v/2-1} e^{-v\lambda_i/2} \quad (24.92)$$

Furthermore, if we define $V_0 = v_0 I$, we can sample v_0 as well. For example, suppose we use a $IG(\delta_1, \delta_2)$ prior for v_0 . The posterior is given by $p(v_0|w) = IG(\delta_1 + 1/2D, \delta_2 + 1/2D \sum_{j=1}^n w_j^2)$. This can be interleaved with the other Gibbs sampling steps, and provides an appealing Bayesian alternative to cross validation for setting the strength of the regularizer.

See Table 24.1 for a summary of various algorithms for fitting probit and logit models. Many of these methods can also be extended to the multinomial logistic regression case. For details, see (Scott 2009; Fruhwirth-

Schnatter and Frühwirth 2010).

24.5.2 Slice sampling

Consider sampling from a univariate, but multimodal, distribution $\tilde{p}(x)$. We can sometimes improve the ability to make large moves by adding an auxiliary variable u . We define the joint distribution as follows:

$$\hat{p}(x, u) = 1/Z_p \text{ if } 0 \leq u \leq \tilde{p}(x) \quad 0 \text{ otherwise} \quad (24.93)$$

where $Z_p = \int p(x) dx$. The marginal distribution over x is given by

$$\hat{p}(x, u) du = \tilde{p}(x) Z_p du = \tilde{p}(x) Z_p = p(x) \quad (24.94)$$

so we can sample from $p(x)$ by sampling from $\hat{p}(x, u)$ and then ignoring u . The full conditionals have the form

$$p(u|x) = U[0, \tilde{p}(x)](u) \quad (24.95)$$

$$p(x|u) = U_A(x) \quad (24.96)$$

where $A = \{x : \tilde{p}(x) \geq u\}$ is the set of points on or above the chosen height u . This corresponds to a slice through the distribution, hence the term slice sampling (Neal 2003a). See Figure 24.14(a).

In practice, it can be difficult to identify the set A . So we can use the following approach: construct an interval $x_{\min} \leq x \leq x_{\max}$ around the current point x_s of some width. We then test to see if each end point lies within the slice. If it does, we keep extending in that direction until it lies outside the slice. This is called stepping out. A candidate value x is then chosen uniformly from this region. If it lies within the slice, it is kept, so $x_s + 1 = x$. Otherwise we shrink the region such that x forms one end and such that the region still contains x_s . Then another sample is drawn. We continue in this way until a sample is accepted.

To apply the method to multivariate distributions, we can sample one extra auxiliary variable for each dimension.

The advantage of slice sampling over Gibbs is that it does not need a specification of the full-conditionals, just the unnormalized joint. The advantage of slice sampling over MH is that it does not need a user-specified proposal distribution (although it does require a specification of the width of the stepping out interval).

Figure 24.14(b) illustrates the algorithm in action on a synthetic 1d problem. Figure 24.15 illustrates its behavior on a slightly harder problem, namely binomial logistic regression. The model has the form

$$y_i \sim \text{Bin}(n_i, \text{logit}(\beta_1 + \beta_2 x_i)) \quad (24.97)$$

We use a vague Gaussian prior for the β 's. Figure 24.15(a)

shows a grid-based approximation to the posterior, and Figure 24.15(b) shows a sample-based approximation. In this example, the grid is faster to compute, but for any problem with more than 2 dimensions, the grid approach is infeasible.

24.5.3 Swendsen Wang

Consider an Ising model of the following form:

$$p(x) = 1/Z e^{f_e(x_e)} \quad (24.98)$$

where $x_e = (x_i, x_j)$ for edge $e = (i, j)$, $x_i \in \{+1, -1\}$, and the edge factor f_e is defined by $e^{-J} e^{-J} e^{-J} e^{-J}$, where J is the edge strength. Gibbs sampling in such models can be slow when J is large in absolute value, because neighboring states can be highly correlated. The Swendsen Wang algorithm (Swendsen and Wang 1987) is a auxiliary variable MCMC sampler which mixes much faster, at least for the case of attractive or ferromagnetic models, with $J > 0$.

Suppose we introduce auxiliary binary variables, one per edge. 5 These are called bond variables, and will be denoted by z . We then define an extended model $p(x, z)$ of the form

$$p(x, z) = 1/Z e^{g_e(x_e, z_e)} \quad (24.99)$$

where $z_e \in \{0, 1\}$, and we define the new factor as follows: $g_e(x_e, z_e = 0) = e^{-J} e^{-J} e^{-J} e^{-J}$, and $g_e(x_e, z_e = 1) = e^{-J} - e^{-J} 0 0 e^{-J} - e^{-J}$. It is clear that $\sum_{z_e=0} g_e(x_e, z_e) = f_e(x_e)$, and hence that $\sum_z p(x, z) = p(x)$. So if we can sample from this extended model, we can just throw away the z samples and get valid x samples from the original distribution.

Fortunately, it is easy to apply Gibbs sampling to this extended model. The full conditional $p(z|x)$ factorizes over the edges, since the bond variables are conditionally independent given the node variables. Furthermore, the full conditional $p(z_e|x_e)$ is simple to compute: if the nodes on either end of the edge are in the same state ($x_i = x_j$), we set the bond z_e to 1 with probability $p = 1 - e^{-2J}$, otherwise we set it to 0. In Figure 24.16 (top right), the bonds that could be turned on (because their corresponding nodes are in the same state) are represented by dotted edges. In Figure 24.16 (bottom right), the bonds that are randomly turned on are represented by solid edges.

To sample $p(x|z)$, we proceed as follows. Find the

connected components defined by the graph induced by the bonds that are turned on. (Note that a connected component may consist of a singleton node.) Pick one of these components uniformly at random. All the nodes in each such component must have the same state, since the off-diagonal terms in the $g_e(x_e, z_e = 1)$ factor are 0. Pick a state ± 1 uniformly at random, and force all the variables in this component to adopt this new state. This is illustrated in Figure 24.16 (bottom left), where the green square denotes the selected connected component, and we choose to force all nodes within it to enter the white state.

The validity of this algorithm is left as an exercise, as is the extension to handle local evidence and non-stationary potentials.

It should be intuitively clear that Swendsen Wang makes much larger moves through the state space than Gibbs sampling. In fact, SW mixes much faster than Gibbs sampling on 2d lattice Ising models for a variety of values of the coupling parameter, provided $J > 0$. More precisely, let the edge strength be parameterized by J/T , where $T > 0$ is a computational temperature. For large T , the nodes are roughly independent, so both methods work equally well.

However, as T approaches a critical temperature T_c , the typical states of the system have very long correlation lengths, and Gibbs sampling takes a very long time to generate independent samples. As the temperature continues to drop, the typical states are either all on or all off. The frequency with which Gibbs sampling moves between these two modes is exponentially small. By contrast, SW mixes rapidly at all temperatures.

Unfortunately, if any of the edge weights are negative, $J < 0$, the system is frustrated, and there are exponentially many modes, even at low temperature. SW does not work very well in this setting, since it tries to force many neighboring variables to have the same state. In fact, computation in this regime is provably hard for any algorithm (Jerrum and Sinclair 1993, 1996).

24.5.4 Hybrid/Hamiltonian MCMC

In this section, we briefly mention a way to perform MCMC sampling for continuous state spaces, for which we can compute the gradient of the (unnormalized) log-posterior. This is the case in neural network models, for

example.

The basic idea is to think of the parameters as a particle in space, and to create auxiliary variables which represent the “momentum” of this particle. We then update this parameter/ momentum pair according to certain rules (see e.g., (Duane et al. 1987; Neal 1993; MacKay 2003; Neal 2010) for details). The resulting method is called hybrid MCMC or Hamiltonian MCMC. The two main parameters that the user must specify are how many leapfrog steps to take when updating the position/ momentum, and how big to make these steps.

Performance can be quite sensitive to these parameters (although see (Hoffman and Gelman 2011) for a recent way to set them automatically). This method can be combined with stochastic gradient descent (Section 8.5.2) in order to handle large datasets, as explained in (Ahn et al. 2012).

Recently, a more powerful extension of this method has been developed, that exploits secondorder gradient information. See (Girolami et al. 2010) for details.

下面我先给出逐句完整译文（不遗漏），再用通俗解释把关键点讲清楚，最后给出要点总结。

逐句完整译文（不遗漏）

24.5 辅助变量 (Auxiliary variable) MCMC

有时，我们可以通过引入虚拟的辅助变量，显著提高采样效率，从而降低原变量之间的相关性。若原变量记作 x ，辅助变量记作 z ，我们要求（对 z 做边缘化后） $\sum_z p(x, z) = p(x)$ （或 $\int p(x, z) dz = p(x)$ ），并且 $p(x, z)$ 比单独的 $p(x)$ 更容易采样。若满足这两个条件，我们就能在扩展后的模型里采样，然后把采到的 z 扔掉，从而恢复来自 $p(x)$ 的样本。下面给出一些例子。

24.5.1 逻辑回归的辅助变量采样

在第 9.4.2 节，我们讨论了 probit 回归的潜变量解释。回忆其形式为

$$z_i = \mathbf{w}^\top \mathbf{x}_i + \varepsilon_i \quad (24.83)$$

$$\varepsilon_i \sim N(0, 1) \quad (24.84)$$

$$y_i = 1 \Leftrightarrow I(z_i \geq 0) \quad (24.85)$$

我们在第 11.4.6 节利用了该表示，用 EM 求 ML 估计。把它改成**辅助变量的吉布斯采样器**很直接（练习 24.6）：因为 $p(\mathbf{w} \mid D)$ 是**高斯**，而 $p(z_i \mid \mathbf{x}_i, y_i, \mathbf{w})$ 是**截断高斯**，两者都易于采样。

现在讨论如何为**逻辑回归**构造辅助变量吉布斯采样器。令 ε_i 服从**Logistic 分布**，其密度为

$$p_{\text{Logistic}}(\varepsilon) = \frac{e^{-\varepsilon}}{(1 + e^{-\varepsilon})^2} \quad (24.86)$$

其均值 $E[\varepsilon] = 0$ ，方差 $\text{var}[\varepsilon] = \pi^2/3$ 。其分布函数为 $F(\varepsilon) = \text{sigm}(\varepsilon)$ ，即 Logistic 函数。由于 $y_i = 1$ 当且仅当 $\mathbf{w}^\top \mathbf{x}_i + \varepsilon > 0$ ，由对称性得

$$p(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}) = \int_{-\infty}^{\infty} f(\varepsilon) d\varepsilon = \int_{-\infty}^{\mathbf{w}^\top \mathbf{x}_i} f(\varepsilon) d\varepsilon = F(\mathbf{w}^\top \mathbf{x}_i) = \text{sigm}(\mathbf{w}^\top \mathbf{x}_i) \quad (24.87)$$

如所需。

我们可以通过在 $p(z \mid \mathbf{w}, D)$ 与 $p(\mathbf{w} \mid z, D)$ 间交替采样来得到一个辅助变量吉布斯采样器。不幸的是，**直接从 $p(\mathbf{w} \mid z, D)$ 采样不可行**。一种办法是设 $\varepsilon_i \sim N(0, \lambda_i)$ ，其中 $\lambda_i = (2\psi_i)^2$ ，并令 $\psi_i \sim \text{KS}$ (Kolmogorov-Smirnov 分布)，随后对 $\mathbf{w}, z, \lambda, \psi$ 采样 (Holmes and Held, 2006)。

更简单的办法是用 **Student-t 分布近似 Logistic** (Albert and Chib, 1993)。具体地，作近似 $\varepsilon_i \sim T(0, 1, \nu)$ ，其中 $\nu \approx 8$ 。此时可用 **Gaussians 的尺度混合**来简化推断：

$$\lambda_i \sim \text{Ga}(\nu/2, \nu/2) \quad (24.88)$$

$$\varepsilon_i \sim N(0, \lambda_i^{-1}) \quad (24.89)$$

$$z_i = \mathbf{w}^\top \mathbf{x}_i + \varepsilon_i \quad (24.90)$$

$$y_i = 1 \mid z_i \Leftrightarrow I(z_i \geq 0) \quad (24.91)$$

此时所有完整条件都有简单形式；细节见练习 24.7。

注意，若取 $\nu = 1$ ，则 $z_i \sim N(\mathbf{w}^\top \mathbf{x}_i, 1)$ ，这与 **probit 回归**等价（见第 9.4 节）。与其在 **probit** 与 **logit** 间二选一，我们可以**直接估计** ν 。它没有便利的共轭先验，但可以在一个有限取值集合上计算其后验：

$$p(\nu \mid \lambda) \propto p(\nu) \prod_{i=1}^N \frac{1}{\Gamma(\nu/2)(\nu/2)^{\nu/2}} \lambda_i^{\nu/2-1} e^{-\nu\lambda_i/2} \quad (24.92)$$

此外，若令 $V_0 = \nu_0 I$ ，我们也可以对 ν_0 采样。例如，若给 ν_0 赋 $IG(\delta_1, \delta_2)$ 先验，则其后验为

$$p(v_0 \mid \mathbf{w}) = \text{IG}\left(\delta_1 + \frac{1}{2}D, \delta_2 + \frac{1}{2} \sum_{j=1}^D w_j^2\right).$$

这可以与其他吉布斯步骤交替进行，为设定正则化强度提供了一个有吸引力的贝叶斯替代（替代交叉验证）。

表 24.1 总结了若干拟合 probit / logit 模型的算法。这些方法多数也能扩展到多项式逻辑回归。细节见 (Scott, 2009; Frühwirth-Schnatter and Frühwirth, 2010)。

24.5.2 片段采样 (Slice sampling)

考虑从一个一维、多峰的未归一化分布 $\tilde{p}(x)$ 采样。我们有时可以通过加入一个辅助变量 u 来提升“大步移动”的能力。定义如下联合分布：

$$\hat{p}(x, u) = \begin{cases} \frac{1}{Z_p}, & 0 \leq u \leq \tilde{p}(x) \\ 0, & \text{否则} \end{cases} \quad (24.93)$$

其中 $Z_p = \int \tilde{p}(x) dx$ 。对 u 做边缘化得

$$\int \hat{p}(x, u) du = \tilde{p}(x) \int_0^{1/Z_p} du = \frac{\tilde{p}(x)}{Z_p} = p(x) \quad (24.94)$$

因此我们可通过从 $\hat{p}(x, u)$ 采样并忽略 u ，来从 $p(x)$ 采样。其完整条件为

$$p(u \mid x) = \text{U}[0, \tilde{p}(x)](u) \quad (24.95)$$

$$p(x \mid u) = \text{U}_A(x) \quad (24.96)$$

其中 $A = \{x : \tilde{p}(x) \geq u\}$ 是在高度 u 处“切片”所覆盖的集合（因此称为 slice sampling；Neal, 2003a）。见图 24.14(a)。

实践中，找集合 A 可能不易，因此可用如下办法：在当前点 x_s 周围构造一个宽度给定的区间 $[x_{\min}, x_{\max}]$ 。测试区间两端是否落在切片内；若是，则在该方向继续扩展，直到超出切片（称为 stepping out）。然后从该区域均匀抽取候选 x' 。若 x' 位于切片内，则接受 ($x_{s+1} = x'$)；否则就收缩该区域（让 x' 成为一端，同时确保区域仍包含 x_s ），再抽一次，如此往复直至接受。

要将该方法用于多维分布，可为每个维度再采一个辅助变量。与吉布斯相比，slice 采样的优点是不需要写出各个完整条件，只要有未归一化联合密度即可；与 MH 相比，它不需要用户自定提议分布（但需要指定 stepping-out 的宽度）。

图 24.14(b) 展示了该算法在一个合成一维问题上的运行。图 24.15 展示了其在稍难的问题（即二项式逻辑回归）上的表现。模型为

$$y_i \sim \text{Bin}(n_i, \text{logit}(\beta_1 + \beta_2 x_i)) \quad (24.97)$$

我们对 β_j 使用一个**宽松高斯先验**。图 24.15(a) 是基于栅格的后验近似，图 24.15(b) 是基于样本的近似。在该例中，栅格更快，但对于维度 > 2 的任何问题，栅格方法都是不可行的。

24.5.3 Swendsen–Wang 算法

考虑如下形式的**伊辛模型**：

$$p(x) = \frac{1}{Z} \prod_e f_e(x_e) \quad (24.98)$$

其中边 $e = (i, j)$ 的变量 $x_e = (x_i, x_j)$ ，且 $x_i \in \{+1, -1\}$ 。边因子 f_e 由矩阵 $\begin{bmatrix} e^J & e^{-J} \\ e^{-J} & e^J \end{bmatrix}$ 定义， J 为边强度。当 $|J|$ 大时，吉布斯在此类模型中会很慢，因为相邻状态高度相关。**Swendsen–Wang (SW) 算法** (Swendsen and Wang, 1987) 是一种**辅助变量 MCMC**，在**吸引/铁磁模型** ($J > 0$) 下混合得更快。

设我们引入**每条边一个的辅助二值变量** (称为 **bond 变量**)，记为 z 。定义扩展模型

$$p(x, z) = \frac{1}{Z} \prod_e g_e(x_e, z_e) \quad (24.99)$$

其中 $z_e \in \{0, 1\}$ ，新因子定义为：当 $z_e = 0$ 时，

$$g_e(x_e, 0) = \begin{bmatrix} e^{-J} & e^{-J} \\ e^{-J} & e^{-J} \end{bmatrix}; \text{ 当 } z_e = 1 \text{ 时，}$$

$$g_e(x_e, 1) = \begin{bmatrix} e^J - e^{-J} & 0 \\ 0 & e^J - e^{-J} \end{bmatrix}.$$

显然 $\sum_{z_e} g_e(x_e, z_e) = f_e(x_e)$ ，因此边缘化 z 得 $\sum_z p(x, z) = p(x)$ 。于是若能从扩展模型采样，把 z 丢掉即可得到原分布的有效 x 样本。

幸运的是，对该扩展模型应用吉布斯很容易。**条件于** x ，各条边的 bond 变量相互独立，故 $p(z | x)$ 在边上分解。且每条边的完整条件 $p(z_e | x_e)$ 很简单：若边两端结点同号 ($x_i = x_j$)，则以概率 $p = 1 - e^{-2J}$ 置 $z_e = 1$ ；否则置 $z_e = 0$ 。图 24.16 (右上) 用虚线表示“可以点亮”的边；图 24.16 (右下) 用实线表示“随机点亮”的 bond。

要从 $p(x | z)$ 采样，按以下步骤：在“点亮”的 bond 所诱导的图上，找其**连通分量** (单点也算一个分量)。**均匀随机选一个分量**；由于 $g_e(x_e, 1)$ 的**非对角项为 0**，该分量中所有结点必须处于**同一状态**。在 ± 1 间**均匀**选一个状态，强制该分量内所有变量都取该状态。图 24.16 (左下) 示意：绿色方块是被选中的连通分量，我们选择把其中所有结点都置为“白色”状态。

该算法的正确性留作练习，如何扩展以处理**局部证据与非平稳势**也留作练习。

直觉上很清楚：与吉布斯相比，SW 在状态空间中迈的是**大步**。事实上，在二维晶格伊辛模型中，只要 $J > 0$ ，在一系列耦合强度下，SW 的混合**远快于**吉布斯。更具体地，令边强度参数化为 J/T ， $T > 0$ 为“计算温度”。当 T 很大时，结点近似独立，两法都好使；当 T 接近**临界温度** T_c 时，系统的典型状态具有很长的相关长度，吉布斯要花很久才能产生独立样本；当 T 继续降低时，典型状态要么“全开”要么“全关”，吉布斯在这两峰之间跳转的频率**指数级小**。相比之下，SW 在所有温度下都**混合快速**。

不幸的是，若有负边权 ($J < 0$)，系统处于**受挫** (frustrated) 状态，即便低温也有**指数多个模态**。SW 在此设定下表现不佳，因为它试图让许多相邻变量同态。事实上，这一 regime 下的计算对任何算法都可证明是困难的 (Jerrum and Sinclair 1993, 1996)。

24.5.4 混合/哈密顿 MCMC (HMC)

本节简述一种用于**连续状态空间且可计算** (未归一化) **对数后验梯度**的 MCMC 方法 (例如神经网络模型)。

基本思想：把参数看作空间中的一个**粒子**，并引入表示该粒子“**动量**”的辅助变量。随后按一定的力学规则更新“位置/动量”对 (见 Duane et al., 1987; Neal, 1993; MacKay, 2003; Neal, 2010)。由此得到的方法称为 **混合 MCMC 或哈密顿 MCMC (HMC)**。用户需指定两个关键参数：**跃蛙 (leapfrog) 步数与步长**；性能对它们**较敏感** (不过见 Hoffman and Gelman, 2011, 有自动设定的新方法)。该法可与**随机梯度下降** (第 8.5.2 节) 结合以处理大数据集 (Ahn et al., 2012)。

最近有更强的扩展，利用了**二阶梯度信息**；见 Girolami et al. (2010) 了解细节。

通俗解释（一步步讲清）

1) “辅助变量”到底有啥用？

有些后验 $p(x)$ 很难直接采而且**相关性强** (走得慢)。我们加个**辅助变量** z ，构造“更大但更好采”的 $p(x, z)$ ，保证“把 z 扔掉还能回到 $p(x)$ ”。这样在 (x, z) 空间里走路可以**更顺、更快**，因为相关性被“打散”了。

2) 逻辑回归的两个套路

- **probit (高斯噪声)**：加潜变量 $z_i = \mathbf{w}^\top \mathbf{x}_i + \varepsilon_i$, $\varepsilon \sim N$ 。于是 \mathbf{w} 的条件分布是高斯、 z_i 的条件分布是截断高斯——**吉布斯超好做**。
- **logit (Logistic 噪声)**：直接做不太行。两个变招：
 1. **Holmes–Held**：把 Logistic 噪声表示成与 KS 分布相关的**方差混合高斯**，连同一堆辅助变量一起抽。

2. Albert–Chib: 用 Student-t 近似 Logistic ($\nu \approx 8$ 很像) , 而 Student-t = 高斯的尺度混合 \Rightarrow 条件分布都变简单了。

小妙招: 把 ν 也当参数估 ($\nu = 1$ 就退化成 probit) , 再把先验方差 ν_0 一起采, 等于贝叶斯地“自己调正则化强度”。

3) Slice 采样 (切片采样) : 不用提议分布!

思路: 在曲线 $\tilde{p}(x)$ 下随便画一条高度 u 的“切片”, 然后在这条“切片”覆盖的 x 区域里均匀抽。做法是:

- 先从 $[0, \tilde{p}(x_s)]$ 抽 u ;
- 再找“切片区域” $A = \{x : \tilde{p}(x) \geq u\}$, 在其中均匀抽新 x' 。

实现上用“stepping out + shrink”: 从当前点往两边把区间撑到切片外, 再在区间里抽; 若抽中在片外, 就把区间向里收。

优点: 只需能算未归一化密度, 不用写完整条件 (比吉布斯省心), 也不用设提议分布 (比 MH 省调参)。

注意: 多维可每一维配一个 u , 但要选好步出宽度。

4) Swendsen–Wang: 让伊辛模型“一步跨好多格”

吉布斯在强耦合 ($|J|$ 大) 时很慢, 因为邻居强相关、一步只改一格。SW 的妙计:

- 给每条边加个 bond (能否“连起来”的指示)。如果两端同号, 以 $1 - e^{-2J}$ 的概率点亮; 否则不亮。
- 点亮后图被分成若干连通块; 一次挑一个块, 把里面所有点一起翻成 +1 或 -1。

这就像把“成片的自旋”一把子翻了, 走得巨快, 尤其在临界温度附近吉布斯会“黏住”的地方, SW 还能飞。但若存在负边权 (反铁磁、受挫), 模式指数多, SW 也难 (理论上任何算法都难)。

5) HMC: 带动量的“物理式”采样

把参数当“粒子”, 引入动量当辅助变量, 按哈密顿力学的轨迹 (用跃蛙积分) 前进, 大步、方向准、接受率高。

要点: 要定步长和步数 (NUTS 可自适应), 还能和随机梯度结合搞大数据。甚至再用二阶信息 (Riemann/Hessian) 更猛。

要点总结

- **核心:** 加“看不见的帮手”(辅助变量) 把难问题变易: 在更大的 (x, z) 空间里采, 最后丢掉 z 仍得 $p(x)$ 。
- **逻辑回归:**
 - probit: 高斯潜变量 \Rightarrow 条件分布简单, 吉布斯易。

- logit: 用 KS-混合或 Student-t 近似 + 尺度混合高斯, 把难度化解; ν 与先验强度 ν_0 还能一起采。
- Slice 采样: 只要未归一化密度, 无需提议、无需完整条件; 靠“切片 + 步出/收缩”在多峰分布上大步走。
- Swendsen-Wang: 为伊辛模型加 bond, 把“整片变量”一起翻, 混合飞快 ($J > 0$ 时); 但受挫系统 ($J < 0$) 对任何算法都难。
- HMC: 引入动量, 用梯度 (甚至二阶) 沿物理轨迹飞奔, 步子大、拒绝少; 参数 (步长/步数) 要调, NUTS可自动化, 能结合随机梯度做大数据。

如果你愿意, 我可以给你做一个**小对比演示**: 同一数据上用 MH、Slice、HMC 的轨迹、接受率、有效样本量 (ESS) 对比, 直观看到“辅助变量/梯度信息”如何提速。

这一节主要在讲什么? 我不太懂

已思考 11s

不慌~这节 (24.5) 讲的是一件事: 加“**辅助变量**”来提速。思路是把原来的难分布 $p(x)$ 扩成更大的、但更好抽样的联合 $p(x, z)$ 。在 (x, z) 空间里采样 (相关性更小、迈得更大步), 最后把 z 扔掉, 样本仍然来自 $p(x)$ 。

下面按小模块说清楚每个招式在干嘛、何时用、优缺点 ↗

大意是什么

- **核心:** 通过引入“看不见的帮手” z , 把“难抽又相关”的后验变成“容易抽、相关性小”的联合; 采完把 z 丢掉仍是对的。
- **好处:** 更快混合 (更快“忘记起点”)、步子更大、有效样本量更高。

四类典型方法 (这节逐一举例)

1. 逻辑/Probit 回归的辅助变量法

- Probit: 设潜变量 $z_i = w^\top x_i + \varepsilon_i$, $\varepsilon_i \sim N(0, 1)$, 观测是 $y_i = \mathbf{1}[z_i \geq 0]$ 。于是
 - $p(w | \cdot)$ 是高斯,
 - $p(z_i | \cdot)$ 是截断高斯,
 ⇒ 吉布斯采样很好做。
- Logit (逻辑回归) : 直接用吉布斯不顺。两招解决:
 - 把 logistic 噪声写成**方差混合高斯** (涉及 KS 分布的混合), 连同额外变量一起采。

- 或用 Student-t 近似 logistic ($\nu \approx 8$) , 而 Student-t 又是高斯的尺度混合 \Rightarrow 条件分布简单、吉布斯可行。还能把 ν 当参数估 ($\nu = 1$ 退化成 probit) , 并把先验强度 ν_0 一起采, 等于贝叶斯自动调正则化。

2. Slice sampling (切片采样)

- 加一个高度变量 u , 在“曲线 $\tilde{p}(x)$ 下”的水平切片里均匀抽 x 。
- 操作: 先抽 $u \sim U[0, \tilde{p}(x_s)]$, 再在集合 $A = \{x : \tilde{p}(x) \geq u\}$ 里抽新 x 。实现用“步出 (stepping out) + 收缩 (shrink) ”。
- 优点: 只要会算未归一化密度, 不需要写完整条件 (比吉布斯省心), 也不需要手动设计提议分布 (比 MH 省调参) 。
- 注意: 要给步出区间设个宽度; 多维可按维度扩展。

3. Swendsen–Wang (伊辛模型的团簇更新)

- 在伊辛模型里给每条边加 bond 辅助变量: 两端同号就以 $1 - e^{-2J}$ 的概率“点亮”。点亮的边把图分成若干连通块。
- 一次选一个连通块, 把里面变量一起翻成 +1 或 -1 \Rightarrow 一步跨很多格, 混合巨快 (对 $J > 0$ 的铁磁情形尤其明显) 。
- 局限: 若有负边权 (受挫、反铁磁), 模态指数多, 任何算法都难, SW 也不灵。

4. HMC (哈密顿/混合 MCMC)

- 给参数加“动量”作辅助变量, 用梯度沿哈密顿动力学轨迹跑 (用跃蛙积分), 步大方向准, 拒绝率低。
- 需要设步长与步数 (NUTS 可自适应); 还能与随机梯度结合做大数据; 有用二阶信息的更强变体 (Riemannian HMC) 。

该怎么选 (小抄)

- 做二分类回归:
 - 喜欢共轭/闭式 \Rightarrow Probit + 潜变量吉布斯。
 - 想要 Logit \Rightarrow Student-t 近似 + 尺度混合 (或 KS 混合) 做吉布斯。
- 多峰一维/低维 \Rightarrow Slice, 少调参、能大步。
- **离散网格/图模型 (像伊辛) **且 $J > 0 \Rightarrow$ Swendsen–Wang, 团簇翻转更快。
- 连续可导高维 (神经网络、广义线性模型等) \Rightarrow HMC/NUTS, 效率高。

一句话版优缺点

- 辅助变量的共同点: 把难题“抬高维度”换成容易的条件更新, 减相关、加速度。
- Slice: 免提议、免完整条件; 要设宽度。

- SW：团簇大步走，遇受挫系统就难。
- HMC：用梯度“物理式”快跑，需调步长/步数，NUTS可省心。

要点总结

- 本节主旨：**引入辅助变量，让采样更快更稳。**
- 逻辑/Probit：用潜变量或尺度混合把 Logit/Probit 都变成**吉布斯友好的**问题； v 、 v_0 可一并贝叶斯估。
- Slice：靠“切片”在复杂/多峰分布中**大步探索**，不需提议分布或完整条件。
- Swendsen–Wang：在图模型里**团簇更新**，对 $J > 0$ 显著提速；对 $J < 0$ 难。
- HMC：连续可导场景的首选之一，**步大、相关低、ESS 高**；NUTS/SG-HMC 等实用化强。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24.6 Annealing methods

Many distributions are multimodal and hence hard to sample from. However, by analogy to the way metals are heated up and then cooled down in order to make the molecules align, we can imagine using a computational temperature parameter to smooth out a distribution, gradually cooling it to recover the original “bumpy” distribution. We first explain this idea in more detail in the context of an algorithm for MAP estimation. We then discuss extensions to the sampling case.

24.6.1 Simulated annealing

Simulated annealing (Kirkpatrick et al. 1983) is a stochastic algorithm that attempts to find the global optimum of a black-box function $f(x)$. It is closely related to the MetropolisHastings algorithm for generating samples from a probability distribution, which we discussed in Section 24.3. SA can be used for both discrete and continuous optimization.

The method is inspired by statistical physics. The key quantity is the Boltzmann distribution, which specifies that the probability of being in any particular state x is given by

$$p(x) \propto \exp(-f(x)/T) \quad (24.100)$$

where $f(x)$ is the “energy” of the system and T is the computational temperature. As the temperature approaches 0 (so the system is cooled), the system spends more and more time in its minimum energy (most probable) state.

Figure 24.17 gives an example of a 2d function at different temperatures. At high temperatures, $T \gg 1$, the surface is approximately flat, and hence it is easy to move around (i.e., to avoid local optima). As the temperature cools, the largest peaks become larger, and the smallest peaks disappear. By cooling slowly enough, it is possible to “track” the largest peak, and thus find the global optimum. This is an example of a continuation method. We can generate an algorithm from this as follows. At each step, sample a new state according to some proposal distribution $x \sim q(\cdot|x_k)$. For real-valued parameters, this is often simply a random walk proposal, $x = x_k + k$, where $k \sim N(0, \Sigma)$. For discrete optimization, other kinds of local moves must be defined.

Having proposed a new state, we compute

$$\alpha = \exp((f(x_k) - f(x_{\text{new}}))/T) \quad (24.101)$$

We then accept the new state (i.e., set $x_{k+1} = x_{\text{new}}$) with probability $\min(1, \alpha)$, otherwise we stay in the current state (i.e., set $x_{k+1} = x_k$). This means that if the new state has lower energy (is more probable), we will definitely accept it, but if it has higher energy (is less probable), we might still accept, depending on the current temperature. Thus the algorithm allows “down-hill” moves in probability space (up-hill in energy space), but less frequently as the temperature drops.

The rate at which the temperature changes over time is called the cooling schedule. It has been shown (Kirkpatrick et al. 1983) that if one cools sufficiently slowly, the algorithm will provably find the global optimum. However, it is not clear what “sufficiently slowly” means. In practice it is common to use an exponential cooling schedule of the following form: $T_k = T_0 C^k$, where T_0 is the initial temperature (often $T_0 \sim 1$) and C is the cooling rate (often $C \sim 0.8$). See Figure 24.18(a) for a plot of this cooling schedule. Cooling too quickly means one can get stuck in a local maximum, but cooling too slowly just wastes time. The best cooling schedule is difficult to determine; this is

one of the main drawbacks of simulated annealing.

Figure 24.18(b) shows an example of simulated annealing applied to the function in Figure 24.17 using a random walk proposal. We see that the method stochastically reduces the energy over time. Figures 24.19 illustrate (a histogram of) samples drawn from the cooled probability distribution over time. We see that most of the samples are concentrated near the global maximum. When the algorithm has converged, we just return the largest value found.

24.6.2 Annealed importance sampling

We now describe a method known as annealed importance sampling (Neal 2001) that combines ideas from simulated annealing and importance sampling in order to draw independent samples from difficult (e.g., multimodal) distributions.

Suppose we want to sample from $p_0(x) \propto f_0(x)$, but we cannot do so easily; for example, this might represent a multimodal posterior. Suppose however that there is an easier distribution which we can sample from, call it $p_n(x) \propto f_n(x)$; for example, this might be the prior. We can now construct a sequence of intermediate distributions than move slowly from p_n to p_0 as follows:

$$f_j(x) = f_0(x)\beta_j f_n(x)^{1-\beta_j} \quad (24.102)$$

where $1 = \beta_0 > \beta_1 > \dots > \beta_n = 0$, where β_j is an inverse temperature. (Contrast this to the scheme used by simulated annealing which has the form $f_j(x) = f_0(x)\beta_j$; this makes it hard to sample from p_n .) Furthermore, suppose we have a series of Markov chains $T_j(x, x')$ (from x to x') which leave each p_j invariant. Given this, we can sample x from p_0 by first sampling a sequence $z = (z_{n-1}, \dots, z_0)$ as follows: sample $z_{n-1} \sim p_n$; sample $z_{n-2} \sim T_{n-1}(z_{n-1}, \cdot)$; ...; sample $z_0 \sim T_1(z_1, \cdot)$. Finally we set $x = z_0$ and give it weight

$$w = f_{n-1}(z_{n-1}) f_{n-2}(z_{n-2}) f_{n-3}(z_{n-3}) \dots f_1(z_1) f_0(z_0) \quad (24.103)$$

This can be shown to be correct by viewing the algorithm as a form of importance sampling in an extended state space $z = (z_0, \dots, z_{n-1})$. Consider the following distribution on this state space:

$$p(z) \propto f(z) = f_0(z_0) \sim T_1(z_0, z_1) \sim T_2(z_1, z_2) \dots \sim T_{n-1}(z_{n-2}, z_{n-1}) \quad (24.104)$$

where \sim is the reversal of T_j :

$$T^* j(z, z') = T_j(z', z) p_j(z') / p_j(z) = T_j(z', z) f_j(z') / f_j(z) \quad (24.105)$$

It is clear that z_1, \dots, z_{n-1} $f(z) = f_0(z_0)$, so we can safely just use the z_0 part of these sequences to recover the original distribution.

Now consider the proposal distribution defined by the algorithm:

$$q(z) \propto g(z) = f_n(z_{n-1}) T_{n-1}(z_{n-1}, z_{n-2}) \cdots T_2(z_2, z_1) T_1(z_1, z_0) \quad (24.106)$$

One can show that the importance weights $w = f(z_0, \dots, z_{n-1}) g(z_0, \dots, z_{n-1})$ are given by Equation 24.103.

24.6.3 Parallel tempering

Another way to combine MCMC and annealing is to run multiple chains in parallel at different temperatures, and allow one chain to sample from another chain at a neighboring temperature. In this way, the high temperature chain can make long distance moves through the state space, and have this influence lower temperature chains. This is known as parallel tempering. See e.g., (Earl and Deem 2005) for details.

下面我先给出逐句完整译文（不遗漏），再用通俗解释把关键点讲清楚，最后给出要点总结。

逐句完整译文（不遗漏）

24.6 退火 (Annealing) 方法

许多分布是多峰的，因此很难采样。不过，类比金属被加热再冷却以使分子重新排列的方式，我们可以想象用一个计算温度参数把分布“抹平/变光滑”，再逐步降温以恢复原本“坑坑洼洼”的分布。我们先在一个用于 MAP 估计的算法背景下更详细地解释这个想法。然后讨论在采样场景下的扩展。

24.6.1 模拟退火 (Simulated annealing)

模拟退火 (Kirkpatrick 等, 1983) 是一种随机算法，尝试为一个黑箱函数 $f(x)$ 找到全局最优。它与我们在 24.3 节讨论过的用于从概率分布生成样本的 Metropolis–Hastings 算法密切相关。SA 可用于离散与连续优化。

该方法受统计物理启发。关键量是玻尔兹曼分布，它规定处于任一特定状态 x 的概率为

$$p(x) \propto \exp(-f(x)/T) \quad (24.100)$$

其中 $f(x)$ 是系统的“能量”， T 是计算温度。当温度趋近 0 (系统被冷却) 时，系统会在其**最小能量 (最可能)**的状态上花越来越多时间。

图 24.17 给出了一个二维函数在不同温度下的例子。高温时 ($T \gg 1$)，表面近似平坦，因此容易移动 (即不容易被局部最优困住)。随着降温，最大的峰变得更大，最小的峰消失。通过**足够缓慢地降温**，有可能“追踪”最大的峰，从而找到全局最优。这是一种**连续化 (continuation) **方法的例子。

我们可以据此生成一个算法：每一步，依据某个提议分布采一个新状态 $x \sim q(\cdot | x_k)$ 。对实值参数，这通常就是随机游走提议 $x = x_k + \epsilon_k$ ，其中 $\epsilon_k \sim N(0, \Sigma)$ 。离散优化则需定义其他类型的局部移动。

提出新状态后，计算

$$\alpha = \exp\left(\frac{f(x) - f(x')}{T}\right) \quad (24.101)$$

随后以 $\min(1, \alpha)$ 的概率**接受**新状态 (即置 $x_{k+1} = x'$)，否则保持在当前状态 ($x_{k+1} = x_k$)。这意味着：如果新状态能量更低 (更可能)，一定接受；若能量更高 (更不可能)，是否接受取决于当前温度。因此该算法允许在**概率空间做“下坡”** (在能量空间“上坡”) 的移动；但随着温度下降，这类移动会**越来越不常见**。

温度随时间变化的速率称为**冷却日程 (cooling schedule)**。已有结果 (Kirkpatrick 等, 1983) 表明：如果降温**足够慢**，该算法在理论上**必然能找到全局最优**。但“足够慢”具体指什么并不清楚。实践中常用如下**指数型**冷却： $T_k = T_0 C^k$ ，其中初始温度 T_0 常取 ~ 1 ，冷却率 C 常取 ~ 0.8 。见图 24.18(a) 的示意。降温太快容易卡在局部最优；太慢则浪费时间。**最佳冷却日程**很难确定，这是模拟退火的主要缺点之一。

图 24.18(b) 展示了对图 24.17 中函数使用随机游走提议进行模拟退火的例子。可以看到，方法会**随机地**让能量随时间降低。图 24.19 展示了 (直方图形式的) 从**降温后的**概率分布随时间抽得样本。多数样本集中在**全局最大**附近。当算法收敛时，我们只需返回找到的**最大值**。

24.6.2 退火重要性采样 (Annealed importance sampling, AIS)

下面描述一种称为**退火重要性采样** (Neal, 2001) 的方法，它将**模拟退火**与**重要性采样**的思想结合起来，以从棘手 (如多峰) 分布中抽取**相互独立**的样本。

设我们想从 $p_0(x) \propto f_0(x)$ 采样，但这很难 (例如它可能是一个多峰的后验)。假设我们可以从一个更容易的分布采样，记为 $p_n(x) \propto f_n(x)$ (例如它可能是先验)。现在构造一列**中间分布**，令它们从 p_n 逐步“移动”到 p_0 ：

$$f_j(x) = f_0(x)^{\beta_j} f_n(x)^{1-\beta_j} \quad (24.102)$$

其中 $1 = \beta_0 > \beta_1 > \dots > \beta_n = 0$, β_j 是逆温度。 (注意与模拟退火的方案对比, 后者用 $f_j(x) = f_0(x)^{\beta_j}$; 那样会让从 p_n 采样变难。) 此外, 假设我们有一系列使各自 p_j 保持不变的马尔可夫链转移核 $T_j(x, x')$ (从 x 到 x')。在此条件下, 我们可以如下采 $x \sim p_0$: 先采一个序列 $z = (z_{n-1}, \dots, z_0)$:

采 $z_{n-1} \sim p_n$; 采 $z_{n-2} \sim T_{n-1}(z_{n-1}, \cdot)$;; 采 $z_0 \sim T_1(z_1, \cdot)$ 。最后置 $x = z_0$, 并赋予如下权重

$$w = \frac{f_{n-1}(z_{n-1})}{f_n(z_{n-1})} \cdot \frac{f_{n-2}(z_{n-2})}{f_{n-1}(z_{n-2})} \cdots \frac{f_1(z_1)}{f_2(z_1)} \cdot \frac{f_0(z_0)}{f_1(z_0)} \quad (24.103)$$

可以把该算法看作在**扩展状态空间** $z = (z_0, \dots, z_{n-1})$ 上做重要性采样, 从而证明其正确性。考虑该状态空间上的分布

$$p(z) \propto f(z) = f_0(z_0) \tilde{T}_1(z_0, z_1) \tilde{T}_2(z_1, z_2) \cdots \tilde{T}_{n-1}(z_{n-2}, z_{n-1}) \quad (24.104)$$

其中 \tilde{T}_j 是 T_j 的**时间反转**:

$$\tilde{T}_j(z, z') = T_j(z', z) \frac{p_j(z')}{p_j(z)} = T_j(z', z) \frac{f_j(z')}{f_j(z)} \quad (24.105)$$

显然对 z_1, \dots, z_{n-1} 积分后 $f(z) = f_0(z_0)$, 因此我们可以只使用这些序列的 z_0 部分来恢复原分布。

再看由算法定义的**提议分布**:

$$q(z) \propto g(z) = f_n(z_{n-1}) T_{n-1}(z_{n-1}, z_{n-2}) \cdots T_2(z_2, z_1) T_1(z_1, z_0) \quad (24.106)$$

可以证明**重要性权重** $w = \frac{f(z_0, \dots, z_{n-1})}{g(z_0, \dots, z_{n-1})}$ 就是式 (24.103)。

24.6.3 并行回火 (Parallel tempering)

把 MCMC 与退火结合的另一种方式是: 在**不同温度下并行**运行多条链, 并允许某条链从**相邻温度**的链中采样 (进行交换)。这样, 高温链能在状态空间中做**远距离移动**, 并将这种探索能力“传递”给低温链。这称为**并行回火**。细节可见 (Earl and Deem, 2005)。

通俗解释 (一步步讲清)

1) 退火在干嘛?

把“难走的地形”先用**高温**“抹平”, 让算法敢到处走、不怕小坑; 再**慢慢降温**, 把地形“还原”, 同时**锁定**到真正的高地/低能位置。这个思路既能用来**找最优 (模拟退火)**, 也能用**来抽样 (AIS、并行回火)**。

2) 模拟退火 (SA) = 优化版 MH

- 把目标函数 $f(x)$ 当作“能量”，构造“温度化”的概率 $p(x) \propto e^{-f(x)/T}$ 。
- 每步：先提议一个新点，再按

$$r = \min(1, e^{(f(x)-f(x'))/T})$$

的概率接受。高温时“上坡（变差）”也常被接受，便于跳出局部最优；低温时更“挑剔”，趋向全局最优。

- 关键难点：**怎么降温？** 太快会卡住；太慢耗时。常用指数降温 $T_k = T_0 C^k$ (如 $C \approx 0.8$) 当经验起点。

3) 退火重要性采样 (AIS) = 架“温度桥”的独立抽样

- 想从难分布 p_0 取样（如多峰后验），但容易从 p_n （如先验）取样。
- 造一串桥接分布 p_j （用逆温度 β_j 混合 f_0 和 f_n ），从易到难逐级过渡。
- 在每个层级用一个能保持 p_j 的 Markov 过渡走一步，把走过的路径赋一个**重要性权重** w ，最后得到对 p_0 的独立、有权样本。
- 直觉：一步步把“先验的粒子”**加热/降温**，逐渐变成“后验的粒子”，中间用权重做校正。

4) 并行回火 (PT) = 多温度多条链，互相“换道”

- 同时跑多条链：**高温链勇于远行**（跨峰）；**低温链精细探索**（峰内）。
- 让相邻温度的链**偶尔交换状态**。这样，低温链能“借”高温链的远行成果，跨越能垒、避免卡顿。
- 常用于后验**多峰**、普通 MH/吉布斯容易困住的情形。

要点总结

- **退火思想**：用“温度”把复杂分布**抹平**便于探索，再**降温**聚焦于真实的峰/低能区。
- **模拟退火 (SA)**：优化用；按温度控制接受“变差”的提议；**冷却日程**是性能关键（难调）。
- **AIS**：在**易→难**的桥接分布序列上行走，并用**重要性权重**校正，得到针对目标分布的**独立样本**。
- **并行回火 (PT)**：多温度链**并行+交换**；高温负责“远跳”，低温负责“精扫”，适合**多峰**难题。
- **实务提示：**
 - SA 找**全局最优**；
 - AIS 取**独立样本**（评估证据、期望等很方便）；
 - PT 让 MCMC 在多峰上**不再黏**。

- 三者都体现一个核心：用温度/桥接来改善探索能力，避免被局部结构困住。

如果你愿意，我可以用一个**玩具多峰函数**给你同时演示 SA（降温轨迹）、AIS（桥接权重分布与有效样本量）和 PT（链间交换与跨峰能力），用图形直观看“温度”是如何帮算法走出泥潭的。

这一节主要在讲什么？我不太懂

已思考 6s

不慌～这节（24.6 “退火方法”）讲的主线就是：**用“温度”帮算法在多峰（好多局部极值）的地形里更好地探索**。先把目标分布/目标函数“加热”变平滑，好走不容易卡；再慢慢降温，回到原来的复杂形状，同时把解/样本“锁”在真正好的区域。

下面抓三件事就懂了 ↗

一句话概括

用温度 T 控制“地形”平不平：**高温平滑、易乱走；低温还原、精细找峰**。这套思路既能找最优解（模拟退火），也能从难分布取样（退火重要性采样、并行回火）。

三个招式，各干啥

1. 模拟退火 (SA) —— 找全局最优

- 把目标函数 $f(x)$ 当“能量”，构造概率 $p(x) \propto e^{-f(x)/T}$ 。
- 每步像 MH 一样：先提议一个新点，再按温度 T 决定是否接受**变差**的点（高温更容易接受，便于跳出局部最优；低温更挑剔）。
- 关键在**冷却日程**（怎么降温）：太快会卡住，太慢浪费时间。常用指数降温 $T_k = T_0 \cdot C^k$ 当经验法。

2. 退火重要性采样 (AIS) —— 从难分布拿“独立样本”

- 有一个**难采样的目标分布** p_0 （比如多峰后验），和一个**容易采的分布** p_n （比如先验）。
- 造一串**中间“温度”分布** $p_n \rightarrow p_{n-1} \rightarrow \dots \rightarrow p_0$ （用逆温度 β 把易→难平滑过渡）。
- 依次在这些分布上走几步，最后得到面向 p_0 的样本，并用一个**权重**做校正；这样能得到**相互独立**的、代表目标分布的样本。

3. 并行回火 (PT) —— 多条不同温度的链互相“换道”

- 同时跑多条 MCMC 链：**高温链**敢“远跳”（跨峰），**低温链**做细致探索。

- 让相邻温度的链偶尔交换状态：低温链能借到高温链的“远跳成果”，不容易黏在某个峰里。

什么时候用哪一个？

- 你要全局优化（只有一个目标函数）：用 模拟退火（SA）。
- 你要从复杂后验抽样，而且希望样本独立：用 AIS。
- 你在做MCMC但分布多峰、链老是卡住：用 **并行回火（PT）** 帮它跨峰。

带走的要点

- 核心思想：温度 = 平滑度。高温好走、低温精准；缓慢降温能避免陷在局部。
- SA：优化版 MH；成败很看冷却日程。
- AIS：搭建“温度桥”，从易到难过渡，配权重拿到面向目标分布的独立样本。
- PT：多温度链 + 交换，高温跨峰、低温精扫，对多峰后验很有效。

如果你愿意，我还能用一个小玩具“多峰函数”画图演示：高温到低温的地形变化、SA 降温轨迹、AIS 的权重与样本、PT 的链间交换是怎么帮助跨峰的。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

24.7 Approximating the marginal likelihood

The marginal likelihood $p(D|M)$ is a key quantity for Bayesian model selection, and is given by

$$p(D|M) = \int p(D|\theta, M)p(\theta|M)d\theta \quad (24.107)$$

Unfortunately, this integral is often intractable to compute, for example if we have non conjugate priors, and/or we have hidden variables. In this section, we briefly discuss some ways to approximate this expression using Monte Carlo. See (Gelman and Meng 1998) for a more extensive review.

24.7.1 The candidate method

There is a simple method for approximating the marginal likelihood known as the Candidate method (Chib 1995).

This exploits the following identity:

$$p(D|M) = \int p(D|\theta, M)p(\theta|M)d\theta \quad (24.108)$$

This holds for any value of θ . Once we have picked some value, we can evaluate $p(D|\theta, M)$ and $p(\theta|M)$ quite easily. If

we have some estimate of the posterior near θ , we can then evaluate the denominator as well. This posterior is often approximated using MCMC.

The flaw with this method is that it relies on the assumption that $p(\theta|D, M)$ has marginalized over all the modes of the posterior, which in practice is rarely possible. Consequently the method can give very inaccurate results in practice (Neal 1998).

24.7.2 Harmonic mean estimate

Newton and Raftery (1994) proposed a simple method for approximating $p(D)$ using the output of MCMC, as follows:

$$\frac{1}{p(D)} \approx \frac{1}{S} \sum_{s=1}^S \frac{1}{p(D|\theta_s)} \quad (24.109)$$

where $\theta_s \sim p(\theta|D)$. This expression is the harmonic mean of the likelihood of the data under each sample. The theoretical correctness of this expression follows from the following identity:

$$\begin{aligned} \int p(D|\theta) p(\theta|D) d\theta &= \int p(D|\theta) p(D|\theta)p(\theta) p(D) d\theta = \int p(D) \\ p(\theta|D) d\theta &= \int p(D) \end{aligned} \quad (24.110)$$

Unfortunately, in practice this method works very poorly. Indeed, Radford Neal called this “the worst Monte Carlo method ever”.⁶ The reason it is so bad is that it depends only on samples drawn from the posterior. But the posterior is often very insensitive to the prior, whereas the marginal likelihood is not. We only mention this method in order to warn against its use. We present a better method below.

24.7.3 Annealed importance sampling

We can use annealed importance sampling (Section 24.6.2) to evaluate a ratio of partition functions. Notice that $Z_0 = f_0(x)dx = f(z)dz$, and $Z_n = f_n(x)dx = g(z)dz$.

Hence

$$\begin{aligned} Z_0/Z_n &= f(z)dz/g(z)dz = f(z)/g(z)dz = \frac{\int f(z)dz}{\int g(z)dz} = \frac{\int f(z)dz}{\int g(z)dz} \\ &\approx \frac{1}{S} \sum_{s=1}^S w_s \end{aligned} \quad (24.111)$$

If f_n is a prior and f_0 is the posterior, we can estimate $Z_n = p(D)$ using the above equation, provided the prior has a known normalization constant Z_0 . This is generally considered the method of choice for evaluating difficult partition functions.

已思考 14s

下面我先给出逐句完整译文（不遗漏），再用通俗解释把难点讲清，最后总结要点方便复习。

逐句完整译文（不遗漏）

24.7 近似边际似然 (marginal likelihood)

用于贝叶斯模型选择的关键量是**边际似然** $p(D \mid M)$, 其定义为

$$p(D \mid M) = \int p(D \mid \theta, M) p(\theta \mid M) d\theta \quad (24.107)$$

不幸的是，这个积分通常难以计算，比如当我们使用非共轭先验，且/或存在隐变量时。本节我们简要讨论一些使用蒙特卡洛近似该表达式的方法。更全面的综述可见 (Gelman and Meng, 1998)。

24.7.1 候选点 (Candidate) 方法

有一种简单的近似边际似然的方法，称为**候选点方法** (Chib, 1995)。它利用如下恒等式：

$$p(D \mid M) = \frac{p(D \mid \theta, M) p(\theta \mid M)}{p(\theta \mid D, M)} \quad (24.108)$$

这对任意给定的 θ 都成立。一旦我们选定某个值，就可以较容易地计算 $p(D \mid \theta, M)$ 和 $p(\theta \mid M)$ 。如果我们对 θ 邻域的后验有某种估计，也可以计算分母。这个后验通常用 MCMC 来近似。

该方法的缺陷在于：它依赖一个假设，即 $p(\theta \mid D, M)$ 已经把后验中的**所有模态都边缘化了**，而这在实践中很少能做到。因此该方法在实践中可能给出非常不准确的结果 (Neal, 1998)。

24.7.2 调和平均估计 (Harmonic mean estimate)

Newton 与 Raftery (1994) 提出了一种用 MCMC 输出近似 $p(D)$ 的简单方法：

$$\frac{1}{p(D)} \approx \frac{1}{S} \sum_{s=1}^S \frac{1}{p(D \mid \theta_s)} \quad (24.109)$$

其中 $\theta_s \sim p(\theta \mid D)$ 。该式就是数据在每个样本下似然的**调和平均**。它的理论正确性来自如下恒等式：

$$\int \frac{1}{p(D \mid \theta)} p(\theta \mid D) d\theta = \int \frac{1}{p(D \mid \theta)} \frac{p(D \mid \theta) p(\theta)}{p(D)} d\theta = \frac{1}{p(D)} \int p(\theta \mid D) \frac{1}{p(D)}$$

不幸的是，该方法在实践中效果很差。事实上，Radford Neal 称其为“**史上最糟糕的蒙特卡洛方法**”。之所以糟糕，是因为它**只依赖后验样本**。而后验往往对先验并不敏感，但**边际似然却高度敏感**。我们提到该法只是为了**提醒不要使用**。下面给出更好的方法。

24.7.3 退火重要性采样 (AIS)

我们可以用**退火重要性采样**（第 24.6.2 节）来评估**配分函数（归一化常数）之比**。注意 $Z_0 = \int f_0(x) dx = \int f(z) dz$, 且 $Z_n = \int f_n(x) dx = \int g(z) dz$ 。因此

$$\frac{Z_0}{Z_n} = \frac{\int f(z) dz}{\int g(z) dz} = \frac{\int \frac{f(z)}{g(z)} g(z) dz}{\int g(z) dz} = E_q\left[\frac{f(z)}{g(z)}\right] \approx \frac{1}{S} \sum_{s=1}^S w_s \quad (24.111)$$

如果把 f_n 取为先验、 f_0 取为后验，那么在先验的归一化常数 Z_0 已知的前提下，就可以用上式估计 $Z_n = p(D)$ 。这通常被认为是评估困难配分函数的**首选方法**。

通俗解释（一步步讲清）

什么是边际似然，为什么难？

- $p(D | M) = \int p(D | \theta, M)p(\theta | M) d\theta$ 是“**模型证据**”：它把参数积分掉，衡量“这个模型总体上能多好地解释数据”。
- 难点：高维积分、非共轭先验、隐变量都会让这积分**没法解析**，数值上也容易崩。

三种近似思路：

1. 候选点 (Chib) 方法

- 公式 (24.108) 把 $p(D | M)$ 写成“(似然×先验) ÷后验密度(在某个 θ 点)”。
- 做法：挑一个“好点”（常是后验众数附近），用 MCMC 近似分母的后验密度值。
- 大坑：如果后验**多峰**而你的 MCMC **没走遍所有模态**，分母会估错 → 整体偏差大。**实务里常不可靠**。

2. 调和平均法

- 用后验样本 θ_s 算 $(\frac{1}{S} \sum 1/p(D | \theta_s))^{-1}$ 。数学上恒等式成立。
- **但方差极大、极不稳**：只要出现少数“似然很小”的样本，整体被它们**主导**，估计会飘到天边。业内共识：**别用**。

3. 退火重要性采样 (AIS)

- 用一串“**温度/桥接**”的中间分布把**易采样**的分布（如先验）逐步过渡到**目标**（如后验），一路记录**权重**。
 - 这些权重的平均，给出**配分函数之比** Z_0/Z_n 。设 f_n 是先验、 f_0 是后验（未归一化），且先验的归一化常数 Z_0 已知，就能解出 $Z_n = p(D)$ 。
 - 直觉：从“先验世界”把粒子一路“降温”搬到“后验世界”，权重负责**纠偏**，最后得到**稳健**的证据估计。实务上被视为**优选方案**（比上面两种靠谱太多）。
-

要点总结

- 边际似然 $p(D | M)$ 是贝叶斯模型选择的核心，但常常**难算**。
- **候选点方法 (Chib)**：用(似然 \times 先验)/后验密度；**多峰时易严重失真**。
- **调和平均法**：理论对、实践崩（方差巨大，受小似然样本支配）；**不推荐**。
- **退火重要性采样 (AIS)**：通过桥接分布计算**配分函数之比**；若先验常数已知，可估 $p(D)$ ；实务中**更可靠**、常为**首选**。
- 实操提示：
 - 用 AIS 时要好好设计**温度序列** (β_j) 和每层的**转移核**，多次运行取平均、用 log-sum-exp 做数值稳定，效果更稳。
 - 一旦得到了 $p(D | M)$ ，就能比较不同模型的**证据**，做贝叶斯模型选择（或算 Bayes 因子）。

如果你愿意，我可以用一个**小玩具例子**（两模型、合成数据）演示：候选点法与调和平均法的数值不稳定对比，以及用 AIS 估计 $p(D)$ 的步骤与结果对照。

这一节主要在讲什么？我不太懂

已思考 7s

不慌～这节 (24.7) 在讲：**怎么用蒙特卡罗近似“边际似然” $p(D | M)$** ，好做**贝叶斯模型选择**（比较模型谁更“解释数据”）。

一句话概括

边际似然是把参数整体“积分掉”的模型证据，常常算不出；本节给了三招近似：**候选点法 (Chib)**、**调和平均法 (不推荐)**、**退火重要性采样 AIS (推荐)**。

核心思路（通俗版）

- **边际似然：** $p(D | M) = \int p(D | \theta, M) p(\theta | M) d\theta$ 。维度高、非共轭、带隐变量 \Rightarrow 直接积分难。
- **我们就靠样本来“估”**这个积分。

1) 候选点法 (Chib)

- **恒等式：** $p(D | M) = \frac{p(D | \theta, M) p(\theta | M)}{p(\theta | D, M)}$ 。
- **做法：**选一处 θ （多半在后验峰附近），用 MCMC 近似分母的后验密度，再把分子分母一除。

- **坑：**后验要是多峰而你的 MCMC 没走遍所有峰，分母就估偏，结果可能很不准。实务里常翻车。

2) 调和平均法 (Harmonic Mean)

- 用后验样本 $\{\theta_s\}$ 计算: $\frac{1}{p(D)} \approx \frac{1}{S} \sum_s \frac{1}{p(D | \theta_s)}$ 。
- 数学上对，但**数值极不稳定** (受极小似然样本支配，方差巨大)。业内共识：**别用**。

3) 退火重要性采样 (AIS)

- 造一串**中间分布**把“容易采”的 p_n (如先验) 平滑过渡到“目标” p_0 (如后验)：

$$f_j(x) = f_0(x)^{\beta_j} f_n(x)^{1-\beta_j}, \quad 1 = \beta_0 > \dots > \beta_n = 0.$$
- 在每一层 p_j 上用能保持其不变的转移 (短 MCMC) 走一步，串起一条“温度路径”，为这条路径算一个**权重** w 。
- 多条路径的权重平均 $\approx Z_0/Z_n$ (配分函数之比)。若**先验的常数** Z_0 已知，就能得 $Z_n = p(D)$ 。
- **优点：**对多峰/难分布稳健，是估证据/配分函数的**首选之一**。

该怎么用 (小抄)

- **要选模型/算 Bayes 因子：**优先 AIS；
- **别用调和平均；**
- Chib 只在**单峰、后验好跑且你能可信估到 $p(\theta | D)$** 时考虑。

实操提示

- AIS 要**设计好温度序列** $\{\beta_j\}$ (从易到难细一点)，每层的**转移核**要能混开；
- 多次独立运行取平均，用 log-sum-exp 做数值稳定；
- 对比模型时，比的是各自的 $p(D | M)$ (或它们的比值 = Bayes 因子)。

要点总结

- 边际似然 = 模型总体“解释力”，但难算；
- Chib：简洁但易受多峰误差影响；
- 调和平均：理论对、实践崩——别用；
- AIS：搭“温度桥”估配分函数之比，**实务推荐**。

基于本对话里我贴过的整章原文，输出：①知识点总结
 (分点+关键词+用途) ②子章节关联 (依赖/对比/递进+关系句，附Mermaid可选) ③本章总结。

已思考 27s

① 知识点总结 (分点 + 关键词 + 用途)

24.1 MCMC 介绍

- 关键点：用**马尔可夫链**构造以目标分布 $p^*(x)$ 为**平稳分布**的随机游走；样本相关；可**做蒙特卡罗积分**。
- 关键词：stationary distribution, random walk, Monte Carlo integration, burn-in。
- 用途：从**高维先验/后验**中取样，估计期望/边际，VI 不便时的通用推断方案。

24.2 Gibbs 采样

- 关键点：逐坐标按**完整条件分布** $p(x_i | x_{-i})$ 轮流抽样；依赖**Markov blanket**；需**烧入**。
- 典型例子：
 - Ising**: $p(x_t = +1 | x_{-t}) = \text{sigm}(2J\eta_t)$ 。
 - GMM**: 对 $\pi, \mu_k, \Sigma_k, z_i$ 给出共轭**完整条件**；有**标签交换**问题；
 - Collapsed Gibbs**: 积分掉参数 (Rao–Blackwell)，方差更小。
 - 分层 GLM**: 学校/学生例；层间共享先验实现**借力**。
- 关键词：full conditional, Markov blanket, label switching, collapsed, blocking。
- 用途：共轭或能写出完整条件的模型；图模型、层级模型、高斯混合等。

24.3 Metropolis–Hastings (MH)

- 关键点：从提议 $q(x' | x)$ 采候选，按
 - 对称: $r = \min(1, \frac{p^*(x')}{p^*(x)})$;
 - 非对称: $r = \min(1, \frac{p^*(x')q(x|x')}{p^*(x)q(x'|x)})$
 接受；**只需未归一化密度**。Gibbs 是 MH 的**特例** (接受率 100%)。
- 提议设计：随机游走；**Hessian 定形** ($2.38^2/D$ 尺度经验)、混合提议、数据驱动提议；**自适应 MCMC**；**可逆跳**处理变维模型。
- 关键词：proposal, acceptance ratio, detailed balance, adaptive MCMC, RJMCMC。
- 用途：难写完整条件、非共轭、逻辑回归等广泛场景。

24.4 速度与准确性

- 关键点：
 - 烧入 (burn-in)：早期样本要丢；
 - 混合时间：与特征值间隙、导通率有关；多峰/瓶颈→慢；
 - 诊断：多链轨迹图、EPSR ($\hat{R} \approx 1$ 更放心)；
 - 相关性：看 ACF，算 ESS；抽稀仅省存储；
 - 链数与长度：通常若干中等长度、多链并行。
- 关键词：mixing time, conductance, trace plot, EPSR(\hat{R}), ACF, ESS。
- 用途：判断是否收敛、样本“含金量”、如何配置运行。

24.5 辅助变量 MCMC

- 关键点：引入**辅助变量** z 使 $p(x, z)$ 易采、相关性小；丢弃 z 还原 $p(x)$ 。
- 招式：
 - 逻辑/Probit：潜变量 & **尺度混合高斯** (Student-t 近似、KS 混合)；
 - Slice：加高度变量 u ，在切片上均匀取样 (stepping-out & shrink)；
 - Swendsen-Wang：Ising 的**团簇翻转**， $J > 0$ 混合快；
 - HMC：加“**动量**”，用梯度/（二阶）信息大步走 (NUTS 可自调)。
- 关键词：auxiliary variable, scale mixture, slice sampling, cluster update, HMC/NUTS。
- 用途：提速、跨峰、降低相关，连续高维 (HMC)、图模型 (SW)、多峰低维 (Slice)。

24.6 退火方法

- 关键点：用“**温度** T ”平滑分布/能量地形，**高温易探索**，**降温锁定解**。
- 三法：
 - **模拟退火 (SA)**：优化用；冷却日程决定成败；
 - AIS：在易→难的**桥接分布**上走并加权，得**独立样本与配分函数比**；
 - **并行回火 (PT)**：多温度链并行与交换，**高温远跳、低温精扫**。
- 关键词：cooling schedule, inverse temperature, bridging distributions, swaps。
- 用途：多峰后验的探索、证据估计、全局优化。

24.7 近似边际似然

- 关键点：
 - **候选点法 (Chib)**：(似然 \times 先验)/后验密度；多峰时易偏；
 - **调和平均**：理论对、方差巨；**不推荐**；
 - AIS：估**配分函数之比** Z_0/Z_n ，若先验常数已知，可得 $p(D)$ 。
- 关键词：model evidence, Bayes factor, partition function ratio。

- 用途：模型选择/比较（证据、Bayes 因子）。

② 子章节关联（依赖 / 对比 / 递进）

- 依赖**
 - 24.1 ⇒ 为 24.2/24.3 提供 MCMC 基本框架（平稳分布、随机游走、烧入）。
 - 24.4 ⇒ 贯穿全章的诊断与效率度量（服务 24.2/24.3/24.5/24.6）。
 - 24.6.2 (AIS) ⇒ 直接支撑 24.7 的边际似然估计。
- 对比**
 - 24.2 (Gibbs) vs 24.3 (MH) : 特例 vs 通用；接受率高但步子小 vs 需调提议但更灵活。
 - 24.5 (Slice/HMC/SW) vs 24.3 (MH) : 都是提议改进/加速；Slice 免提议，HMC 用梯度，SW 做团簇大步。
 - 24.6 (退火) vs 24.4 (直接采样与诊断) : 退火是改善探索的策略；24.4 教你何时算“混好”。
- 递进**
 - 24.2/24.3 基础采样器 → 24.5 加速/结构化改进 → 24.6 温度法跨峰/独立抽样 → 24.7 用 AIS 做证据估计。

Mermaid (可选)

```
flowchart TD A[24.1 MCMC简介] --> B[24.2 Gibbs] A --> C[24.3 MH] B --> D[24.4 速度&准确性] C --> D B --> E[24.5 辅助变量MCMC] C --> E E --> F[24.6 退火方法] C --> F F --> G[24.7 边际似然近似(AIS)] D -.诊断/度量适用于所有.-> B D -.诊断/度量适用于所有.-> C D -.诊断/度量适用于所有.-> E D -.诊断/度量适用于所有.-> F
```

- 关系句示例：
 - "Gibbs ⊂ MH: 把提议设为完整条件, MH 接受率=1。"
 - "AIS→证据: 退火桥接 + 重要性权重 ⇒ 配分函数之比 ⇒ 边际似然。"
 - "PT 辅助 MH/Gibbs: 高温链跨峰, 低温链精扫, 交换缓解黏滞。"

③ 本章总结（实践小抄）

- 目标**: 设计一条能有效探索的链, 使样本长期来自目标分布; 据此做积分/预测/模型选择。
- 方法层级**

1. **基础采样**: Gibbs (能写完整条件时) , MH (通用, 调提议) 。
2. **提速增强**: 辅助变量 (Slice、SW) 、**梯度法 (HMC/NUTS)** 。
3. **温度与桥接**: SA (优化) 、 PT (跨峰) 、 AIS (独立样本/证据) 。
4. **评估与选择**: 用 EPSR/ESS/ACF/trace 检查收敛与效率; 用 AIS 估证据 (避免调和平均; Chib 法谨慎用) 。

- **常见坑**

- 过早用样本 (**未烧入**) → 误结论;
- **多峰仅用朴素 MH/Gibbs** → 混不开;
- 一味追求**高接受率** → 探索差;
- **标签交换直接平均参数** (混合模型) → 无意义;
- **用调和平均估证据** → 极不稳定。

- **选型指南**

- 共轭/可分解: Gibbs/Collapsed Gibbs;
- 连续可导高维: HMC/NUTS;
- 多峰: PT 或 AIS;
- 图模型强耦合: Swendsen–Wang;
- 仅需全局最优: **模拟退火**;
- 要比较模型: AIS 估 $p(D | M)$, 算 Bayes 因子。

- **运行与诊断**

- 多链启动 (初值分散), 看**轨迹重叠**; $\hat{R} \leq 1.1$ 作为经验阈;
- 关注 ESS 而非总样本数; 必要时**并行回火/阻塞更新**;
- 记录**调参** (步长、提议协方差、温度表), 用 pilot run 调到接受率 ~25–40% (随机游走类) 。

一句话收尾: **本章给你一整套“如何抽、如何快、如何判、如何比”的工具箱——从 Gibbs/MH 到 HMC/Slice/SW, 再到 PT/AIS 与证据估计, 配上收敛诊断与实务守则, 足以覆盖常见的贝叶斯推断与模型选择任务。**

请用通俗易懂的话总结本章, 使我能够理解

已思考 15s

好的！用最接地气的话，把第 24 章（MCMC 推断）捋成一张“心智地图”给你——看完能说得清：**MCMC是干嘛的、有哪些招、怎么跑得快、怎么判断好不好、什么时候用哪一招、以及怎么做模型比较。**

这章到底在解决什么问题？

我们常要从**很复杂的分布**（比如贝叶斯后验）里拿样本，用来算期望、做预测、选模型。可直接抽样/积分基本做不到。

MCMC 的主意是：造一条“会乱走但不瞎走”的**马尔可夫链**，长期在某个状态停留的时间，**正比于**目标分布在那儿的概率。于是跑久了、丢掉开头一段“热身”（burn-in），就能用样本做各种估计。

算法全家桶（各有性格）

把分布想成一张“地形图”（高处=概率大、低处=概率小）：

1. Gibbs 采样：

一次只动一个坐标，按“该坐标的条件分布”更新。

- 优点：有共轭/能写出条件分布时，**简单好用**；还能“折叠（collapsed）”掉参数来降方差。
- 缺点：**步子小**，变量强相关或多峰时容易“挪不开”。混合模型还有**标签交换**要注意（别直接平均“第k簇”的参数，去问“i和j在同一簇吗？”这类**不依赖标签**的问题）。

2. Metropolis–Hastings (MH)：

先“提议”新点，再按一个**接受概率**决定跳不跳。**只需要未归一化密度**（常数项都不用）。

- 优点：**通用**，Gibbs 其实是它的特例。
- 关键：**提议分布**要调好——太小不走、太大老被拒。经验上接受率 25%–40% 比较高效；也可以用**Hessian定形状**、**自适应**、**混合提议**、**数据驱动提议**。
- 变体：**可逆跳（RJMCMC）**能在“模型维度可变”时跳转（比如簇数未知）。

3. 辅助变量家族（提速神器）：

给系统加“看不见的帮手”，把难抽的 $p(x)$ 变成更好抽的 $p(x, z)$ ，采完把 z 扔掉。

- **Slice (切片采样)**：加高度 u ，在“切片”区域里均匀抽，不用自己设计提议分布。

- Swendsen–Wang (图模型/伊辛)：把连在一起的一大片变量成片翻转，一步跨很远（铁磁 $J > 0$ 特别好用）。
- HMC (哈密顿) /NUTS：给参数加“动量”，沿“物理轨迹”大步走；需要梯度，适合连续可导高维，相关性低、有效样本量高。

4. 退火方法（用温度帮你跨峰）：

- 模拟退火 (SA)：做优化用。高温敢“上坡”，不怕局部最优；慢慢降温锁定全局。
 - 并行回火 (PT)：多条不同温度的链并行，相邻温度交换，高温负责“远跳”、低温负责“精扫”。
 - 退火重要性采样 (AIS)：搭“温度桥”从易分布走到难分布，带权重校正，能拿到独立样本，还能算配分函数之比（下文用来估证据）。
-

跑得快与跑得准：怎么判断、怎么改进

- 烧入 (burn-in)：开头一段扔掉，别太早下结论（Potts 模型的例子就会误判）。
 - 混合 (mixing)：多峰/有“瓶颈”就慢（导通率低）。这时用 PT、SW、HMC 等更合适。
 - 收敛诊断：
 - 多条链、轨迹图看是否“堆成一团”；
 - \hat{R} (EPSR) ≈ 1 更放心；
 - ACF (自相关) 衰得快更好；
 - ESS (有效样本量) 比“总样本数”更重要；
 - 抽稀只省存储，不提效率。
 - 链数与长度：常用 3 条中等长度，各自丢前半段；从模态初始化可减少烧入。
 - 调参：随机游走类把接受率调到 25–40%；HMC 用 NUTS 省心；PT 的温度梯度要平滑。
-

模型选择：怎么估“证据”（边际似然）

我们要比较模型 M 的“整体解释力”，需要 $p(D | M)$ 。这常常很难算。

- 别用：调和平均（不稳定，被极端样本拖垮）。
- 谨慎用：Chib 候选点（多峰时容易偏）。
- 推荐：AIS——用桥接分布估配分函数之比，先验常数已知时能得到 $p(D)$ ，做 Bayes 因子很稳。

一张“选型小抄”

- **有共轭/条件分布好写** → Gibbs / Collapsed Gibbs / Block Gibbs
- **通用、无梯度** → MH (调好提议；必要时 PT)
- **连续可导高维** → HMC / NUTS
- **一维/低维多峰** → Slice
- **图模型强相关 (如伊辛 $J > 0$)** → Swendsen–Wang
- **卡在局部峰** → 并行回火 (PT)
- **只想找全局最优** → 模拟退火 (SA)
- **要比较模型证据** → AIS (别用调和平均)

最后给你一套“实操流程”

1. **写清目标**: 能算出 (未归一化的) 对数后验即可。
2. **选算法**: 按上面小抄定一招或组合几招。
3. **初始化**: 多条链、初值尽量分散；能从模态起更好。
4. **调节**: 提议协方差/步长/温度表先做 pilot run 调到状态不错。
5. **跑与监控**: 看轨迹、 \hat{R} 、ESS、ACF；必要时用 PT/HMC/阻塞更新提速。
6. **取样与汇总**: 丢 burn-in，计算均值/区间；混合模型注意**标签交换** (用标签不变的量)。
7. **模型比较**: 用 AIS 估 $p(D | M)$ 做 Bayes 因子。

一句话收尾：

MCMC 就是“造路→乱走→久而准”的技术。本章教你：基础路怎么造 (Gibbs/MH)、走得更快的捷径 (Slice/HMC/SW/退火/回火)、怎么判断走得好不好 (\hat{R} 、ESS、ACF)，以及走完后怎么**公平比较模型** (用 AIS 估证据)。有了这套工具箱，绝大多数贝叶斯推断问题都能落地。