

DS 102 Discussion 5

Wednesday, Sep 29, 2021

The past few lectures have looked at how to perform Bayesian inference using Markov Chain Monte Carlo sampling methods. These methods involve drawing samples using a Markov Chain, whose steady-state distribution is a specified target distribution. Since the goal of Bayesian inference is to get the posterior distribution of the parameters given the data, $\mathbb{P}(\theta | X)$, MCMC algorithms can be used to draw samples from the posterior distribution. Often times, the posterior is difficult to derive in closed form, so MCMC methods are an efficient way of approximating it.

1. Gibbs Sampling for Gamma-Poisson model

When the dimension of the parameters is large, sampling from the posterior over *all* the parameters θ is also often difficult. The main insight behind Gibbs sampling is that it can be much easier to sample the posterior over just a *single* parameter, $\mathbb{P}(\theta_i | X, \theta_{-i})$ (where we use the index $-i$ to mean all indices except for i). Gibbs sampling then iterates through each parameter θ_i and samples from $\mathbb{P}(\theta_i | X, \theta_{-i})$. This loop is repeated, each time conditioning on the newly sampled values. Iterating through each parameter θ_i and sampling from $\mathbb{P}(\theta_i | X, \theta_{-i})$ is not the same thing as sampling from $\mathbb{P}(\theta | X)$. However, the good news is that given enough iterations, the former converges to the latter.

Consider the hierarchical Bayes model where

$$\begin{aligned}\beta &\sim \text{Gamma}(m, \alpha) \\ \theta_i | \beta &\sim \text{Gamma}(k, \beta), \quad i = 1, \dots, n \\ X_i | \theta_i &\sim \text{Pois}(\theta_i), \quad i = 1, \dots, n,\end{aligned}$$

where the θ_i are independent of each other and the X_i are independent of each other. The β and θ_i are unknown parameters, and m , α , and k are fixed and known.

We'd like to infer the parameters β and the θ from the data X . That is, we'd like to sample from the posterior distribution $\mathbb{P}(\beta, \theta | X)$ using Gibbs sampling. This entails deriving the posterior of each parameter, conditioned on the data and all the other parameters.

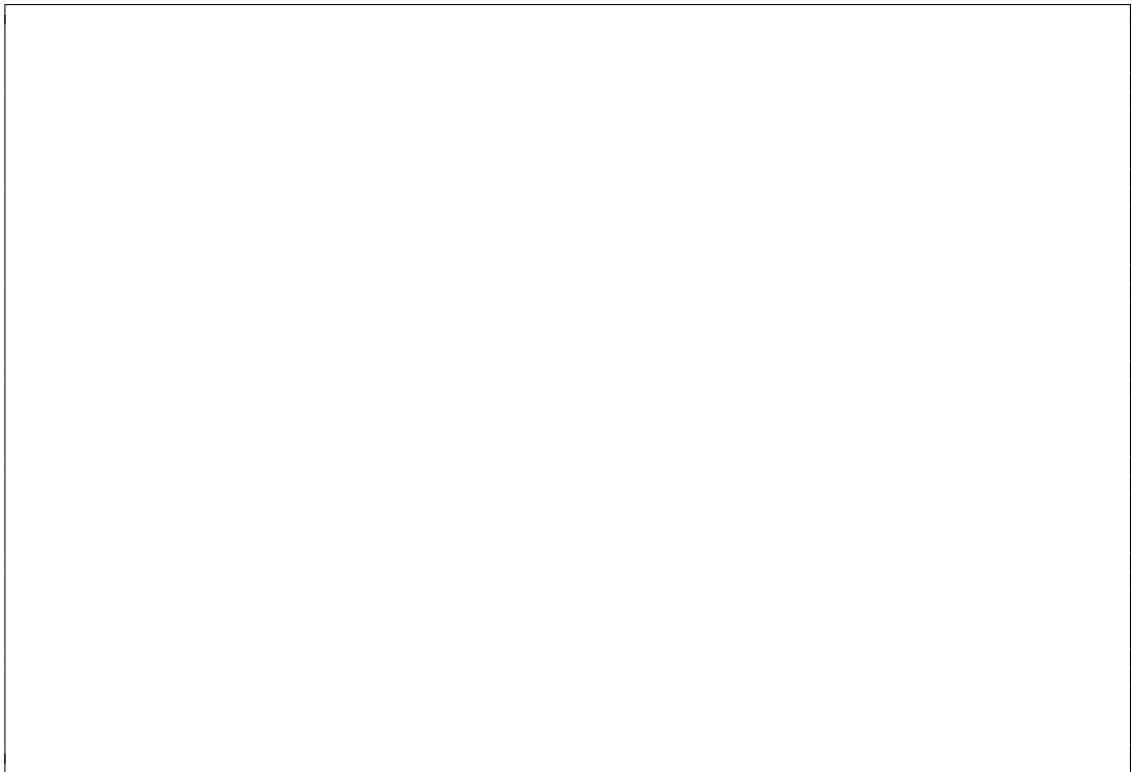
(a) *Drawing the graphical model*

Draw a graphical model which best represents the specified Gamma-Poisson model.



(b) *Finding the conditional distribution of β*

Let's start with β . Derive $\mathbb{P}(\beta \mid \theta_1, \dots, \theta_n, X_1, \dots, X_n)$.

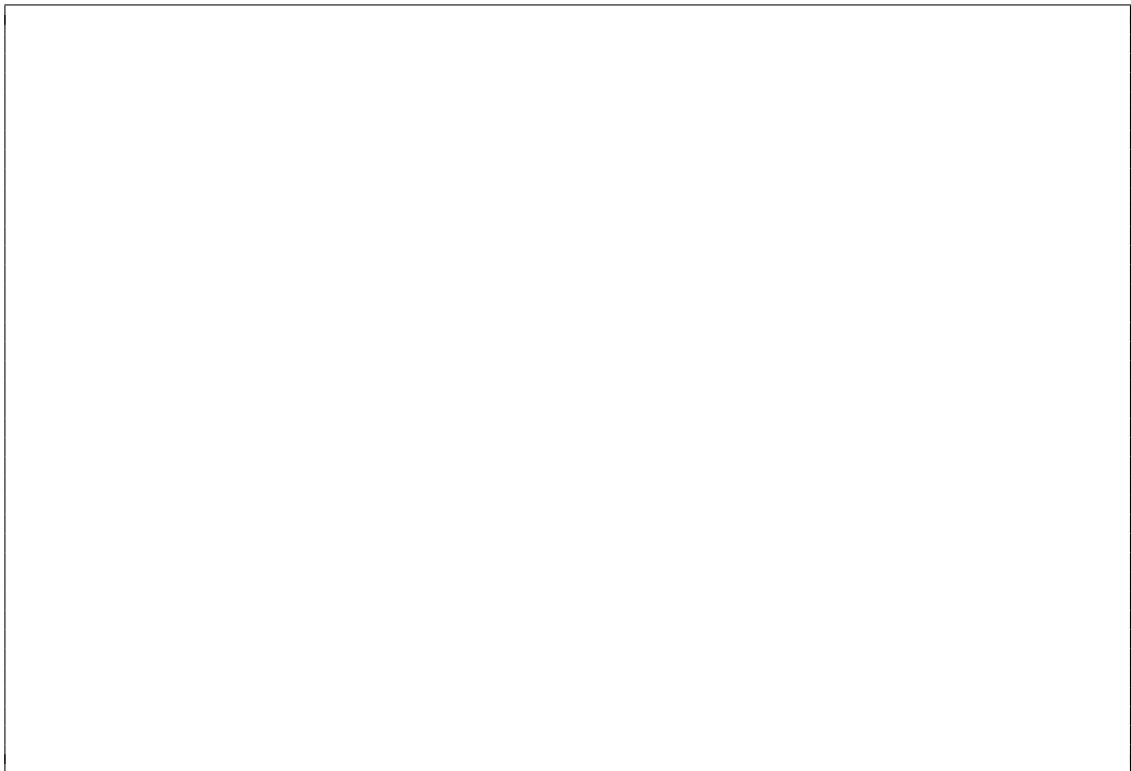


(c) *Finding the conditional distribution of θ_i*

Next, we'll look at each θ_i . Derive $\mathbb{P}(\theta_i \mid \beta, \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_n, X_1, \dots, X_n)$



(d) Using the posteriors you derived in the last two parts, write out the algorithm for the Gibbs sampler.



2. Metropolis-Hastings

This problem proves properties of the **Metropolis-Hastings Algorithm**.

Recall that the goal of MH was to draw samples from a distribution $p(x)$. The algorithm assumes we can compute $p(x)$ up to a normalizing constant via $f(x)$, and that we have a proposal distribution $g(x, \cdot)$. The steps are:

- Propose the next state y according to the distribution $g(x, \cdot)$.
- Accept the proposal with probability

$$A(x, y) = \min \left(1, \frac{f(y) g(y, x)}{f(x) g(x, y)} \right).$$

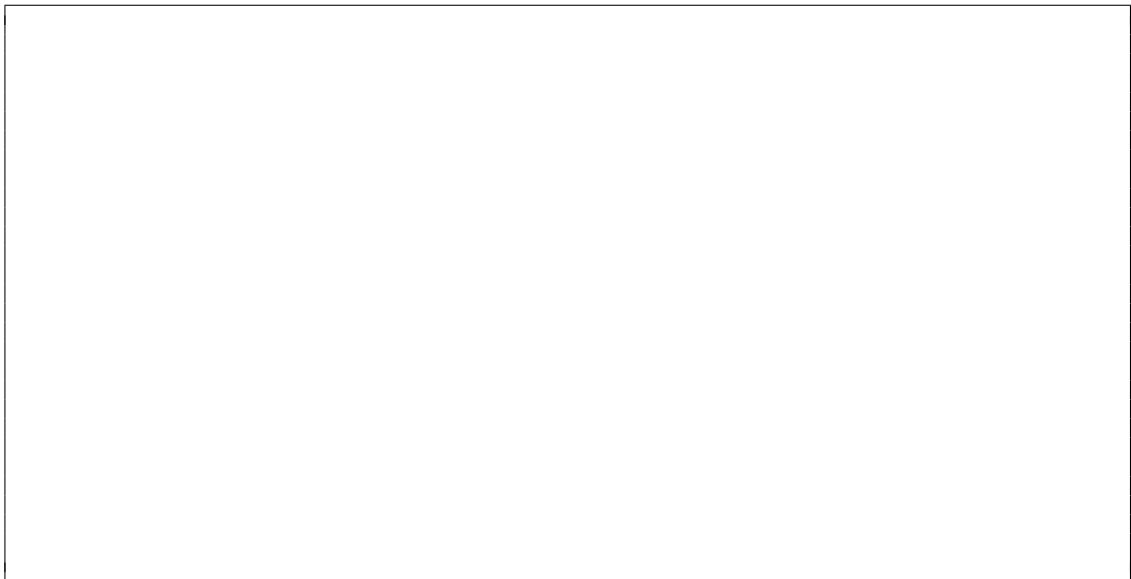
- If the proposal is accepted, then move the chain to y ; otherwise, stay at x .

The key to showing why Metropolis-Hastings works is to look at the **detailed balance equations**. Suppose we have a finite irreducible Markov chain on a state space \mathcal{X} with transition matrix P . If there exists a distribution π on \mathcal{X} such that for all $x, y \in \mathcal{X}$,

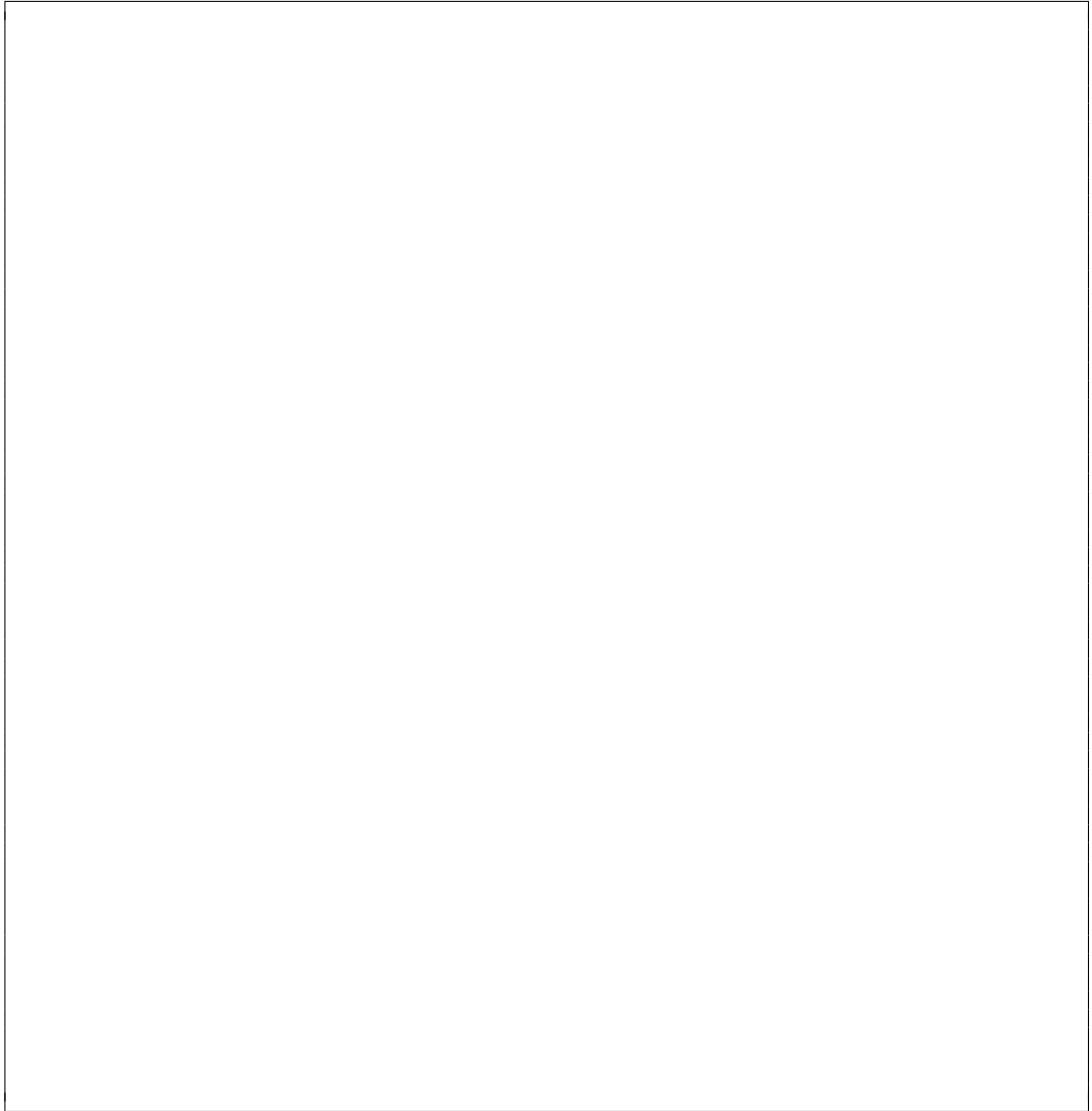
$$\pi(x)P(x, y) = \pi(y)P(y, x),$$

then π is a stationary distribution of the chain (i.e. $\pi P = \pi$).

- (a) For the Metropolis-Hastings chain, what is $P(x, y)$ in this case? For simplicity, assume $x \neq y$.



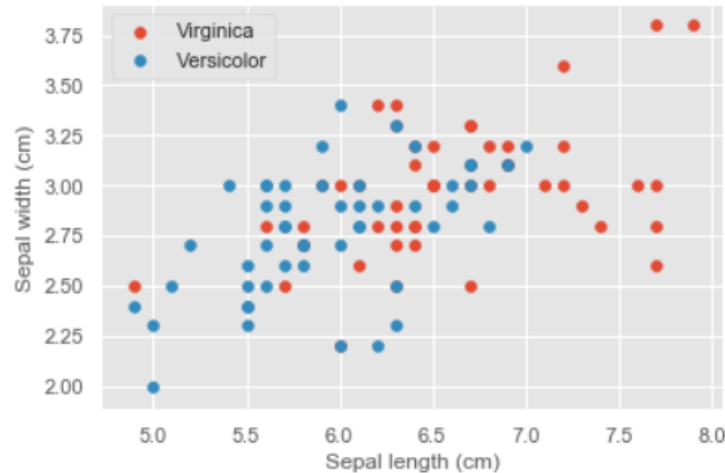
- (b) Show $p(x)$, our target distribution, satisfies the detailed balance equations with $P(x, y)$, and therefore is the stationary distribution of the chain.



3. Interpreting the Logistic Model

In this problem, we fit a logistic regression model on a subset of the famous [iris dataset](#). We have 100 samples of iris flowers, and measure their sepal length, sepal width, petal length and petal width. The response labels are whether they belong to the *Virginica* species (1) or the *Versicolor* species (0).

Let's say we fit a Logistic regression model to predict the iris species, using only the sepal features. Then, our data is represented in the following plot:



(a) *Interpreting a coefficient of a logistic model*

Suppose after fitting the aforementioned logistic regression model, you observe the following output:

```

=====
Generalized Linear Model Regression Results
=====
Dep. Variable:          species    No. Observations:      100
Model:                  GLM        Df Residuals:          97
Model Family:          Binomial   Df Model:               2
Link Function:         logit      Scale:                  1.0000
Method:                IRLS       Log-Likelihood:        -55.163
Date:                  Mon, 22 Feb 2021   Deviance:               110.33
Time:                  00:47:22     Pearson chi2:           100.
No. Iterations:        4
Covariance Type:      nonrobust
=====

```

	coef	std err	z	P> z	[0.025	0.975]
const	-13.0460	3.097	-4.212	0.000	-19.117	-6.975
sepal_width	0.4047	0.863	0.469	0.639	-1.286	2.096
sepal_length	1.9024	0.517	3.680	0.000	0.889	2.916

Assuming that the model is correct, write in one sentence an interpretation for the logistic model with respect to sepal length. What happens to the interpretation if the model is misspecified?

Hint: Recall that the logistic model is

$$\log\left(\frac{p}{1-p}\right) = x^T \beta$$

where p is the probability of the sample with feature vector x being in class 1. The quantity on the left hand side is called the log odds ratio.

(b) *Comparing Goodness-of-Fit*

We now build another logistic model which additionally includes petal width as a feature. You are presented with the following summary output:

```

=====
Generalized Linear Model Regression Results
=====
Dep. Variable:          species    No. Observations:          100
Model:                 GLM        Df Residuals:              96
Model Family:          Binomial    Df Model:                   3
Link Function:         logit       Scale:                      1.0000
Method:                IRLS        Log-Likelihood:            -12.951
Date:                  Mon, 22 Feb 2021  Deviance:                   25.902
Time:                  00:47:22      Pearson chi2:               32.6
No. Iterations:        8
Covariance Type:      nonrobust
=====

```

	coef	std err	z	P> z	[0.025	0.975]
const	-20.2873	8.055	-2.519	0.012	-36.075	-4.499
sepal_width	-4.8233	2.097	-2.300	0.021	-8.933	-0.714
sepal_length	1.2951	1.089	1.189	0.234	-0.839	3.430
petal_width	15.9227	3.981	4.000	0.000	8.121	23.725

Which model has a better fit? How can you tell?

(c) *Selecting a Model*

It could be the case that for the some data, we can have two very different models, Model A and Model B, both achieving very good fit. In this case, which should we use to interpret the data generating process?

