STAT 547: Bayesian Workflow

Charles C. Margossian
University of British Columbia
Winter 2026
https://charlesm93.github.io/stat_547/

DRAFT

1 Why Bayesian Workflow?

1.1 Review of Bayesian modeling and Bayesian inference

What is a Bayesian model ?
$\rightarrow$ A joint distribution over model parameters θ and observations y,

$$p(\theta, y) = \underbrace{p(\theta)}_{\text{prior}} \underbrace{p(y \mid \theta)}_{\text{likelihood}}. \tag{1}$$

The prior encodes knowledge (broadly defined) about the parameters before observing the data and the likelihood tells us how to simulate data given a fixed θ.

Bayesian inference reverse-engineers the data generating process: given observations y, what are plausible values of θ? In a Bayesian setting, all inferential conclusions follow from the *posterior*, obtained via Bayes' rule,

$$p(\theta \mid y) = \frac{p(\theta)p(y \mid \theta)}{p(y)} = \frac{p(\theta, y)}{\int p(\theta, y) \mathrm{d}\theta}. \tag{2}$$

More generally, for any quantity of interest $f(\theta)$, want to learn $p(f(\theta) \mid y)$ and we can also consider quantities which stochastically depend on θ.

Key benefits of Bayesian inference:

(i) The prior encodes expert knowledge and serves as a regularization tool.

(ii) The posterior provides principled ways to quantify uncertainty.

Question: Which properties of the posterior do we care about?

Examples of applications of Bayesian modeling:

- Epidemiology: "Estimation of SARS-CoV-2 mortality during the early stages of an epidemic: a modeling study in Hubei, China and six regions in Europe" [Hauser et al., 2020]
- Pharmacokinetics: "Bayesian aggregation of average data: An application in drug development" [Weber et al., 2018]
- Cosmology: "Listening to the noise: Blind Denoising with Gibbs Diffusion" [Heurtel-Depeiges et al., 2024]
- Deep Learning: "Auto-Encoding Variational Bayes" [Kingma and Welling, 2014]

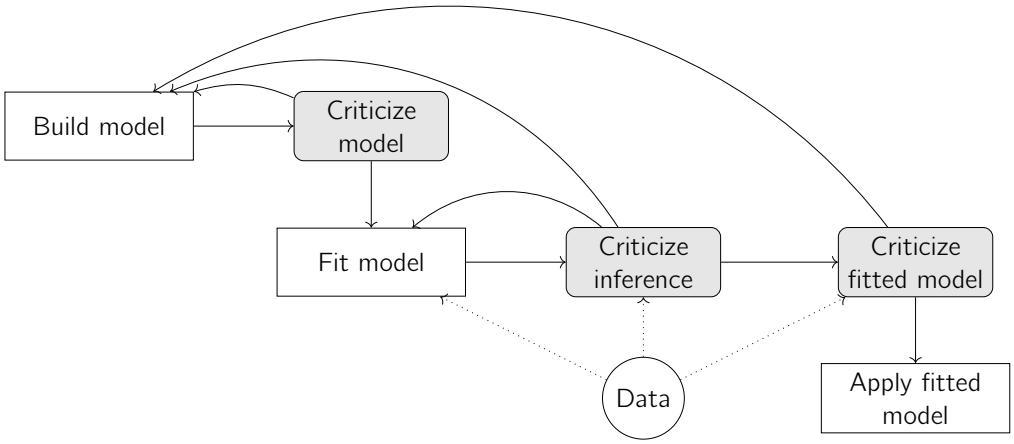

Figure 1: Model development as an iterative process. *Fitting the model is only one step in the broader workflow we use to analyze data. Figure borrowed from Grinsztajn et al. [2021].*

1.2 Going beyond: Bayesian workflow

Key questions:

(i) How do we build a model $p(\theta, y)$?

(ii) Given a model, how do we approximate $p(\theta \mid y)$?

(i) is *modeling* (scientific + statistical) and (ii) is *inference*.

Workflow is the iterative process of building, fitting and criticizing models. Typically an analysis involves a model development phase, understanding the limitations of that model, building an improved model, comparing it to other models... It's a tangled process!

Box's loop suggests an iterative process (Figure 1).

Example (SEIR model). The model we used in Hauser et al. [2020] is the 15^{th} "model iteration". Grinsztajn et al. [2021] discuss earlier models, why they failed and how to improve them. For example:

- Predictions are far-off $\rightarrow$ add a reporting rate parameter.
- Skewed predictions $\rightarrow$ add incubation period.
- ODE solver is too slow $\rightarrow$ adjust parameterization of ODE (then model runs in 2 hours instead of 3 days!)
- ...

What tools do we need to support workflow?

(i) An expressive probabilistic programming language that let's us do model composition.

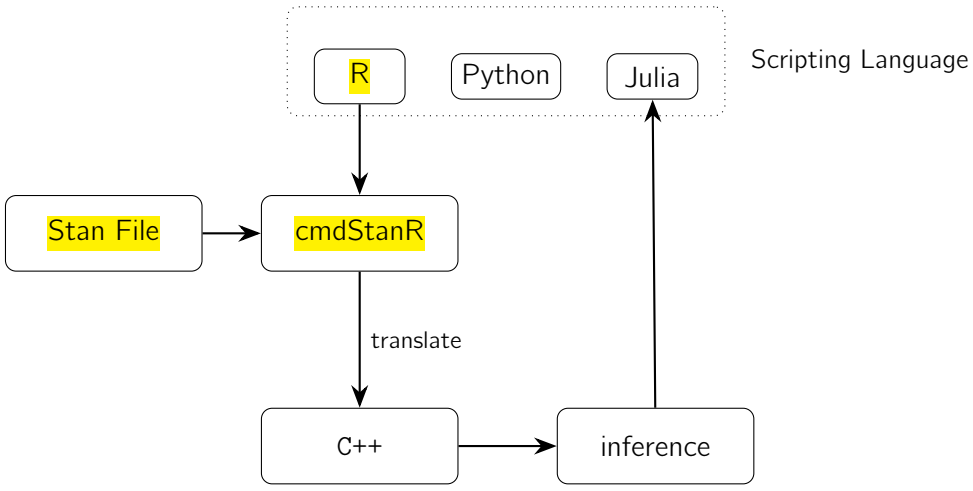

Figure 2: Code workflow when using Stan. *The Bayesian model is specified in a Stan file. Manipulating the data, calling Stan's inference, and analyzing the output of Stan's inference are all done in a scripting language. In this course, we'll use R with the package cmdStanR. Under the hood, the Stan file is translated into a C++ file, which is then compiled and executed when doing inference.*

(ii) Black box inference algorithms that can readily fit bespoke models without requiring extensive tuning effort from users.

(iii) Reliable checks to criticize the inference and the trained model.

1.3 Probabilistic programming languages

Examples: Stan, Pigeons.jl, PyMC, Turing, PyTorch, TensorFlow Probability

Highlighted languages have developers here at UBC!

In this course, we'll focus on Stan, although the concepts we learn apply more broadly.

The two primary components of Stan are:

(i) A flexible language to specify a model, i.e. $p(\theta, y)$.

(ii) Several black box gradient-based inference algorithms which return approximate samples from $p(\theta \mid y)$, e.g. Markov chain Monte Carlo, variational inference.

One thing that makes Stan easy to use is that, in general, the user only needs to specify their model. The gradients are calculated automatically via *automatic differentiation* and Stan provides several "off-the-shelf" inference algorithms. When Stan came out, its two main innovations were an extensive automatic differentiation library for probabilistic modeling and a self-tuning Hamiltonian Monte Carlo algorithm.

While Stan provides its own language for specifying a model, it needs to be interfaced with a scripting language (Figure 2).

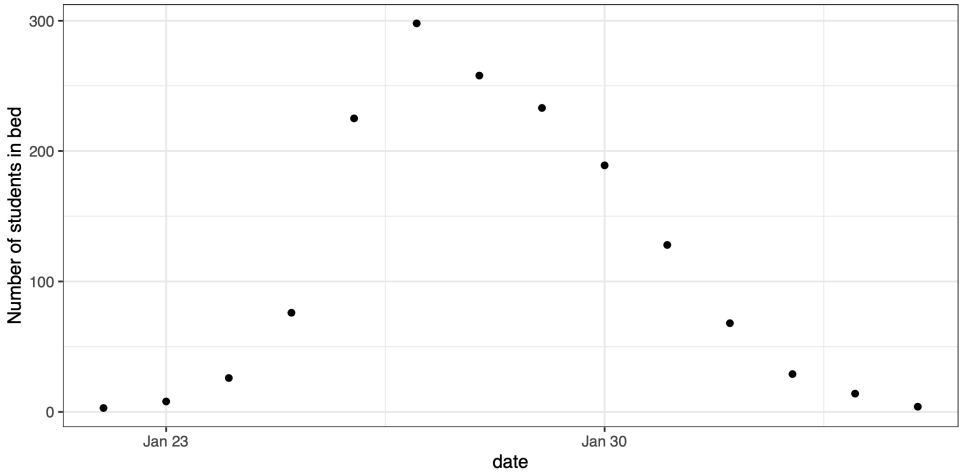

Figure 3: 1978 Influenza outbreak in a British boarding school

Figure 4: *Susceptible-Infected-Recovered model*

1.4 Example: Influenza Outbreak

We're going to analyze data from a 1978 influenza outbreak at a British boarding school (Figure 3). Our goal is to understand the dynamics of the disease and, for example, predict the number of future cases, calculate the recovery time, and calculate R_0.

1.4.1 Scientific model

First we introduce a scientific model of the disease: the Susceptible-Infected-Recovered (SIR) model (Figure 4). This model treats the population as continuous. Each "element" of the population is either susceptible, infected (and infectious) or recovered.

Let N be the total population and $S(t)$, $I(t)$, and $R(t)$ the amount of susceptible, infected and recovered elements at time t. Then the dynamic of the disease are modeled using a system of ordinary differential equations (ODEs):

$$S'(t) = -\beta \frac{S(t)I(t)}{N} \quad (3)$$

$$I'(t) = \beta \frac{S(t)I(t)}{N} - \gamma I(t) \quad (4)$$

$$R'(t) = \gamma I(t), \quad (5)$$

with:

- β: the transmission rate,
- γ: the recovery rate.

In addition, we have an initial condition $S(t_0) = N - 1$, $I(t_0) = 1$, $R(t_0) = 0$ where t_0 is the time at the beginning of the infection. We also have some derived quantities of interest:

- $T = 1/\gamma$: the recovery time
- $R_0 = \beta/\gamma$: how many individual does one person infect.

This scientific model does not tell us how the data is generated. Indeed, none of the quantities in it are observed—and they will need to be infered.

1.4.2 Observational model

Our observations are the number of students in bed $y(t)$) at time t, which is an indirect probe of the number of infected individuals.

Option 1. Poisson likelihood with rate $\lambda(t) = I(t)$. Then $\mathbb{E}y(t) = I(t)$ and $\text{Var}y(t) = I(t)$.

Option 2. Negative binomial with mean $\mu(t) = I(t)$ and overdispersion parameter ϕ. Here $\mathbb{E}y(t) = I(t)$ and $\text{Var}y(t) = I(t) + I^2(t)/\phi$.

Question: what other observational models might we consider?

1.4.3 Prior model

Under option 1, we only have two model parameters: $\beta \in \mathbb{R}^+$ and $\gamma \in \mathbb{R}^+$. We usually have some knowledge of what values these model parameters might take, based on an understanding of the underlying scientific phenomenon and/or other observations on the same disease or similar diseases. This knowledge can be detailed or it might be something like an order of magnitude (for example, we don't expect patients will take 100 years to recover from the disease).

Translating this understanding into a prior model is an active area of research. Let's look at some example:

- $p(\beta) = \text{normal}^+(2, 1) \implies \beta > 0$ and $P(\beta < 4) = 0.975$.
- $p(\gamma) = \text{normal}^+(0.4, 0.5) \implies \gamma > 0$ and $P(\gamma < 1) = 0.9$ (equivalently $P(T > 1) = 0.9$).

Can reason about the prior by seeing what they imply on other quantities of interest such as the recovery time T.

1.4.4 Writing and fitting the joint model in Stan

We now have all the ingredients to write a model in Stan: that is, we've specified a joint distribution over observations and model parameters, which defines a Bayesian model.

Before we start coding, let's pause briefly and ask some fundamental questions:

- Can we trust this model?

  When building our model, we made several choices which can be reasonably contested. For example, it is not correct to treat the number of infected individuals $I(t)$ as a continuous

variable. As George Box said, "all models are wrong but some of them are useful." It falls upon us to identify the ways in which the model is useful and the ways in which the model is wrong. This is at the heart of workflow.

- Can we trust the prior?

  Building priors is often challenging and something practitioners struggle with. People often criticize Bayesian modeling on the basis that we can influence our inference by tweaking the prior. This is true; but it is just as true that we can influence our result by tweaking the likelihood.

  It's good to be skeptical of the prior, in the sense that it's good to be skeptical of the model. On the other hand, being skeptical of the prior without questioning the likelihood is a double-standard. We'll study ways to check how wrong and how useful different components of the model are. But first, we need to talk about fitting the model.

Coding demo: Write the SIR model with Poisson likelihood.

Remark: Stan does not distinguish between prior and likelihood.

With cmdStanR, we can now:

(i) translate the model from Stan to C++ and compile it using `cmdstan_model()`.

(i) Run Stan's default inference engine with `$sample()`.

The output of Stan is approximate samples from the posterior, which are summarized in table.

variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk
gamma	0.476	0.476	0.0110	0.0108	0.459	0.495	1.00	2362
beta	1.69	1.69	0.0149	0.0146	1.67	1.71	1.00	2794
R0	3.55	3.55	0.0786	0.0766	3.42	3.68	1.00	2962
T	2.10	2.10	0.0484	0.0474	2.02	2.18	1.00	2362

Let's focus on the first row for the parameter γ. The first six columns are statistical summaries of the marginal posterior $p(\gamma \mid y)$. The next tow columns tell us about the quality of the inference and whether it can be trusted. Our goal in the next section will be to understand how these diagnostics are computed and how to interpret them.

References

L. Grinsztajn, E. Semenova, C. C. Margossian, and J. Riou. Bayesian workflow for disease transmission modeling in stan. *Statistics in Medicine*, 40(27):6209–6234, 2021. doi: https://doi.org/10.1002/sim.9164.

A. Hauser, M. J. Counotte, C. C. Margossian, G. Konstantinoudis, N. Low, C. L. Althaus, and

J. Riou. Estimation of sars-cov-2 mortality during the early stages of an epidemic: a modeling study in hubei, china and six regions in europe. *PLOS Medicine*, 17(7), 2020.

F. Heurtel-Depeiges, C. C. Margossian, R. Ohana, and B. Régaldo-Saint Blancard. Listening to the noise: Blind denoising with gibbs diffusion. *International Conference on Machine Learning*, PMLR 235:18284–18304, 2024.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR) 2014*, 2014.

S. Weber, A. Gelman, D. Lee, M. Betancourt, A. Vehtari, and A. Racine-Poon. Bayesian aggregation of average data: An application in drug development. *Annals of Applied Statistics*, 12 (3):1583–1604, 2018. doi: 10.1214/17-AOAS1122.