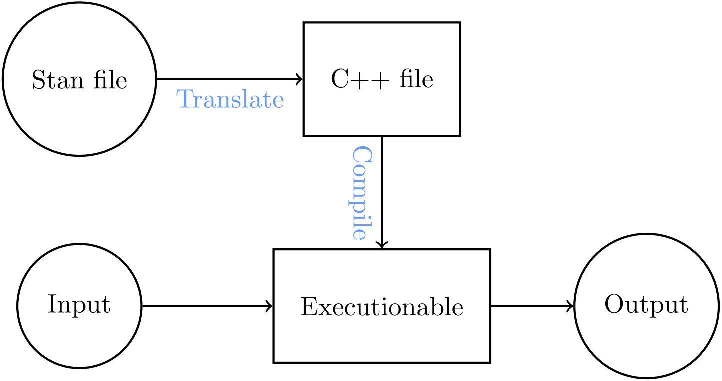
III
Stan

- Stan is an expressive language for joint distributions.
- It automatically computes derivatives.
- It automatically performs inference algorithms.

How Stan works



How Stan works

- The Stan file specifies the joint distribution

$$p(\theta, y) = p(y|\theta)p(\theta) \propto p(\theta \mid y)$$

- The input includes:
  - the data, y
  - tuning parameters for the algorithm
- The output can include:
  - an approximate sample from the posterior distribution
  - summaries of the run which can help us diagnose problems.

Inference algorithms in Stan

- Hamiltonian Monte Carlo (HMC)
- No-U Turn Sampler (NUTS)
- Automatic differentiation variational inference (ADVI)
- Pathfinder Variational Inference
- ...

We can manage the Stan file, the input, and the output using a scripting language, such as:

- R
- Python
- Julia
- The command line
- . . .

Example 1: linear regression

The data generating process is:

$$p(y \mid \theta) = \text{Normal}\left(\beta x, \sigma\right)$$

Our goal is to estimate $\theta = (\beta, \sigma)$, based on the observation $z = (x, y)$ and prior knowledge we have of β and σ.

- `data/linear.data.r`

Example: Bayesian linear regression

As a prior, we use:

- $\beta \sim \text{Normal}(2.0, 1.0)$
- $\sigma \sim \text{Gamma}(1.0, 1.0)$

which encode information from previously observed data.

Writing the Stan file

We need a statement that specifies the log joint distribution.
Recall:

$$p(\theta, y) = p(y \mid \theta)p(\theta)$$

Then:

$$\log p(\theta, y) = \log p(y \mid \theta) + \log p(\theta)$$

Writing the Stan file

Stan retains certain C++ features:

- Variables need to be declared.
- Each statement must end with a semi-colon.

For example:

```
real x;
```

Writing the Stan file

A Stan program is divided into coding blocks:

- data
- parameter
- model

Writing the Stan file

```
data {
 Declare the data that will be given as an input.
}

parameters {
 Declare the parameters we want to sample.
}

model {
 Compute the log joint distribution.
}
```

Writing the Stan file

```
model {
 target += normal_lpdf(y | beta * x, sigma);

 // or equivalently

 y ~ normal(beta * x, sigma);
}
```

Writing the Stan file

Live demo.

Convergence diagnostic

Are the chains still biased by their initializations?

Proposition: Start each chain at a different location and check that they all converge to the same distribution. Look at:

- the trace plots and the density plots to compare estimates from each chain.
- the $\widehat{R}$ statistic.

The $\widehat{R}$ statistic,

$$\widehat{R} = \frac{\text{standard deviation across all chains}}{\text{standard deviation within chain}}.$$

The $\widehat{R}$ statistic,

$$\widehat{R} = \frac{\text{standard deviation across all chains}}{\text{standard deviation within chain}}.$$

- If the chains are sampling from the same target, expect $\widehat{R} \approx 1$.
- If the chains are disagreement, $\widehat{R} \gg 1$.

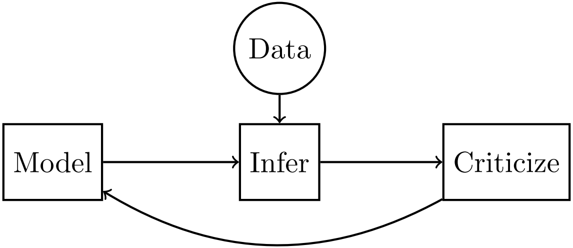
The $\widehat{R}$ statistic,

$$\widehat{R} = \frac{\text{standard deviation across all chains}}{\text{standard deviation within chain}}.$$

- If the chains are sampling from the same target, expect $\widehat{R} \approx 1$.
- If the chains are disagreement, $\widehat{R} \gg 1$.
- What quantity does $\widehat{R}$ measure and how close to 1 should it be?
  - [Vehtari et al., 2021] propose checking that $\widehat{R} \leq 1.01$.
  - [Moins et al., 2022] examine the property of $\widehat{R}$ for stationary chains.
  - [Margossian et al., 2022] examine $\widehat{R}$ for non-stationary chains, and connect $\widehat{R}$ to a measure of bias decay.

Posterior predictive checks

- Recall Box's loop.
- Does our model accurately describe the data?



Posterior predictive checks

Given our posterior distribution for θ, what kind of data, y_{pred}, do we generate?

Proposition:
Each time we draw a sample, $\theta^{(i)} = (\beta^{(i)}, \sigma^{(i)})$, we will also simulate data, according to:

$$y_{\text{pred}}^{(i)} \sim \text{Normal}\left(x\beta^{(i)}, \sigma^{(i)}\right)$$

Posterior predictive checks

To do this, we will use the `generated quantities` block.

Live demo.

Improving the model

- The ppc suggest our model can improve with an intercept parameter.
- *Exercise: repeat the above procedure, but this time add an intercept parameter* β_0.

General resources to use Stan

- The Stan user manual
- The Stan book (https://mc-stan.org/docs/stan-users-guide/index.html)
- The Stan forum (http://discourse.mc-stan.org/)

Parallel chains

- Each chain is completely independent and can be run on a different core.

```
fit <- mod$sample(file = "model/linear.stan",
                  data = data, chains = 4,
                  init = init,
                  parallel_chains = 4)
```

References I

[Margossian et al., 2022] Margossian, C. C., Hoffman, M. D., Sountsov, P., Riou-Durand, L., Vehtari, A., and Gelman, A. (2022).
Nested $\widehat{R}$: Assessing the convergence of markov chain monte carlo when running many short chains.
Preprint. arXiv:2110.13017.

[Moins et al., 2022] Moins, T., Arbel, J., Dutfoy, A., and Girard, S. (2022).
On the use of a local $\widehat{R}$ to improve MCMC convergence diagnostic.
arXiv:2205.06694.

[Vehtari et al., 2021] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021).
Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion).
Bayesian Analysis, 16:667–718.