Applied Probabilistic Programming & Bayesian Machine Learning

Optimizing Stan Code

MIT IAP 2017 Max W. Shen

In general, Stan runs best when inference chains explore primarily on the unit scale, without hard boundaries, and when parameters are uncorrelated. Certain coding tricks can improve the efficiency of Stan coding.

A big topic that arises often in practice is collinearity/correlation between parameters - see section 3 in the course material regarding this topic.

This document acts as a "light summary" of the topics covered in Chapter 22 of the Stan reference.

1 Vectorization

In Stan, matrix and vector operations are more efficient than using for-loops. Nearly all commonly used functions in Stan support vectorization, so it can be worth your time to ensure your parameters or data are in the correct data type to enable vectorization.

Commonly, vectors are used:

```
vector[N] alpha;
```

However, arrays are also useful, especially for distributions like the binomial that require integer inputs.

```
int alpha[N];
```

2 Input Standardization

Stan runs most efficiently when input has roughly zero sample mean and unit sample variance. Data X can be converted into z-scores by:

$$\text{z-score}(x) = \frac{x - \text{mean}(X)}{\text{std}(X)} \tag{1}$$

3 Non-centered and Centered Parameterization

Non-centered parameterization is more efficient when there's little data, while centered parameterization is more efficient when there's a lot of data.

3.1 Centered Parameterization

```
parameters {
  real mu_beta;
  real<lower=0> sigma_beta;
  vector[K] beta;
  ...
model {
  beta ~ normal(mu_beta, sigma_beta);
  ...
```

3.2 Non-centered Parameterization

```
parameters {
  vector[K] beta_raw;
  ...
transformed parameters {
  vector[K] beta;
  // implies: beta ~ normal(mu_beta, sigma_beta)
  beta = mu_beta + sigma_beta * beta_raw;
model {
  beta_raw ~ normal(0, 1);
  ...
```

4 Covariance Matrices

It can be useful in practice to work with the Cholesky decomposition of a covariance matrix.

5 Bibliography and Additional Resources

The Stan reference is available here: http://mc-stan.org/documentation/

Chapter 22 discusses optimizing Stan code for efficiency in greater depth.