Hamiltonian Monte Carlo (HMC) Algorithm and Stan

Stat 341 — Spring 2017

April 2017

King Monty

King Markov has a brother named Monty. King Monty's Kingdom is along a river. There are no towns, per se, people just build their homes here and there along the river. King Monty's royal boat must cruise up and down the river spending more time where there is higher population density. His advisor suggests that they speed up the boat when the population is becoming less dense and slow down the boat when the population becomes more dense. This only requires them to know *how fast the population density is currently changing*, but nothing else.

King Monty and Stan

Stan uses a more complicated algorithm than the Metropolis algorithm, generically called Hamiltonian Monte Carlo (or hybrid Monte Carlo – HMC in either case). This algorithm uses the derivative of (somthing proportional to) the (log) posterior.

HMC Big ideas

- Uses a physics model: Basically simulate a sliding puck on a landscape that is proportional to the log of the posterior.
- Steepness of the log-posterior determines how far the puck slides before turning around.
- Works only for continuous parameter spaces – we need (partial) derivatives of the log-posterior.
- Tuning parameters include mass of puck and force applied to get it moving.

Stan, the implementation that we will use

- **adds a layer of automatic tuning** to make it easier to use
- the initial portion of the chains are used not only to "get the process rolling" but also to figure out how to set the tuning parameters.

Stan requires checking some diagnostics

HMC is much more efficient for certain types of models (especially models with lots of paramters), but doesn't work in all cases. We will need to learn how to check whether the things appear to be working.

Advantages to Stan over Quadratic Approximation (i.e. map())

- We no longer need to assume that the posterior distributions are multivariate normal.
- This was OK for the linear models we have been studying, but there are many modeling situations where this is not a reasonable assumption.
- Highly efficient for many models
- Includes automatic tuning, so tuning is less of an issue.

Getting Started with Stan

Fitting (simple) models with Stan is easy

We can use `map2stan()` which use the same syntax as `map()`. But we need to to two additional things:

- Precompute any variable transformations and store them in new variables.
- Remove any missing data from the data frame – even if it is in variables that the model doesn't use.

`map2stan()` doesn't do everything Stan can do, but it will get us started quickly.

See Chapter 8 code for examples using Stan.