Speeding up Bayesian computations

Now conversions can take quite some time. Imagine running Markov chain for tens or hundreds of thousands of iterations. And you're looking at days or even weeks of running, running them all. Because of that, researchers working within the Bayesian paradigm have tried to come up with ways of speeding up computations or reducing execution time.

In the slides, I go over a couple of methods that can be used to speed up computations. One is in integrating our with c++ as you may know, c++ is very fast programming language. However, running c++ or learning c++ from scratch, can be quite challenging endeavor. RCPP is a package that allows us to use chunks of c++ code and call it from R with relatively minimal c++ knowledge. So I include a series of codes in which I compare the performance of pure standard R with against RCPP. For this, for instance, estimating a logic model. And you will see that convergence takes almost six times as much when we estimate a logit model in Rcompared to the estimation in RCPP.

In addition, another way to further speed up computation is using cluster computing, or high-performance computing. You can think of typically, most research universities in the UK have some sort of cluster in the in Exeter we have ISCA. Pretty much. most universities have cluster computing services, you can also use Amazon cloud. And the key advantage of using cloud computing is parallelization.

So MCMC problems are what's called embarrassingly parallel meaning, imagine you have you're running a model with three parallel chains Well, it's natural to send rather than running them all sequentially, meaning running the first chain waiting until the first chain is finished, and then running the second chain and then subsequently, you can see mechanically send different chains of the of your MCMC programme to different cores.

Now you'll need cluster computing for parallelization. There are packages in R that can be used to paralyse tasks including MCMC chains within your computer. The key advantage of using cluster computing is that just said that the volume or the number of jobs that can be sent the same time. So what I mean by this is, suppose you have a model with you're running three parallel chains to estimate the model. Well, the typical desktop computer has eight cores, which means that you can efficiently paralyse at most two jobs, my jobs mean models you're estimating say well, with cloud computing, you typically have hundreds of cores available. So you can simultaneously estimate 10s or hundreds of models. In fact, I include in in this in the accompanying materials, I include code for a logic model using RCPP estimated using cloud computing. And what we find is that essentially running a model in a logic model. Using RCPP in a cluster computing speeds up takes essentially 13 times more when you estimate the model using R than we do when you estimate the model of using RCPP plus cluster computing. So the efficiency gains in terms of shortening the execution time can be sizable. So, I include a series of a couple of codes I so you can familiarise yourself with running MCMC within classic computing.