The NIMBLE Environment for Statistical Computing

Claudia Wehrhahn1, Abel Rodriguez2, and Christopher Paciorek2

1 Applied Mathematics and Statistics, UC Santa Cruz, USA.
2 Department of Statistics, UC Berkeley, USA.

Introduction

NIMBLE is an open source system within the statistical R software environment. It is a flexible system that allows users to define high-level hierarchical models and select and/or define algorithms for model sampling at each level of the model. Additionally, computations are speed up through C++ compilations. Bayesian non-parametric (BNP) models are a very flexible approach for statistical modeling, yet their, usually difficult, implementation requires the development of a statistical software allowing more users to enjoy the richness of the BNP modeling approach. NIMBLE can be downloaded from https://r-nimble.org. Several examples, the manual, links for questions, and much more, are available from the web page!

The NIMBLE environment involves three components:

• A declarative language for statistical model specification using BUGS language from R. You can write your model using BUGS code and turn it into a model object that can be used for any algorithm provided by NIMBLE or your own. Also, BUGS code is extended to include multiple parametrizations.

• A new domain-specific language for programming computational statistical algorithms. You can write your own algorithm using nimbleFunction() or use one from the NIMBLE library. Algorithms provided by NIMBLE can be customized.

• A compiler that generates C++ code and compiles it. You can use this compiler for speeding up computations in the BUGS model, self-defined algorithm, or any numerical computation, without knowing any C programming!

NIMBLE: BNP modeling

NIMBLE provides support for Bayesian non-parametric (BNP) mixture modeling. The current implementation provides support for hierarchical specifications involving Dirichlet process (DP) mixtures. A DP mixture model takes the form

\[ y_i \sim \text{DP}(\alpha, G_0) \]

As an example, consider a machine assembling hard drives, which sometimes end up being defective. The time (in days) between two consecutive defective hard drives are recorded, in the log scale. The aim is to “find” the density function that describes the time between two consecutive defective hard drives are found.

NIMBLE: model sampling

• Standard code lines for sampling posterior distributions: mConf<--configureMCMC(model); mCMC<--buildMCMC(mConf); CsMCMC<--compileMCMC(mCMC), CsMCMC<--run(1000).

• Specialized algorithms for BNP models: CRP_concentration_sampler: alpha, CRP_sampler: c[1:35]; conjugate_dbeta_dcat sampler: v[1].

• Samplers can be customized at any level of the model. For instance, modelConf$removeSamplers('mutilde') and modelConf$addSampler('mutilde', type='RW_block').

• User-defined samplers can be defined and can be included in the MCMC steps.