1 Introduction

Markov chain Monte Carlo (MCMC) has become a core computational method for many statistical analyses. These include routine Bayesian analyses, but also hybrid algorithms that use MCMC as one component, such as Monte Carlo Expectation Maximization (MCEM; Caffo, Jank, and Jones, 2005) or data cloning (Lele, Dennis, and Lutscher, 2007). Nevertheless, the automated generation of black-box MCMC algorithms, as occurs in generally available software, does not necessarily result in efficient MCMC sampling. Analysts are thereby accustomed to MCMC run times measured in minutes, hours or even days for large hierarchical models. Computation time is frequently the limiting factor, either limiting the range of models considered, or limiting the potential for performing diagnostics and comparisons using methods such as bootstrapping (Efron and Tibshirani, 1994), cross validation (Gneiting and Raftery, 2007), or calibration of posterior predictive p-values (Hjort, Dahl, and Steinbakk, 2006), among others. Therefore, any widely applicable improvements to MCMC performance may greatly improve the practical analyses of large hierarchical models.

Among the many MCMC sampling algorithms developed to improve MCMC efficiency, one of the most basic approaches has been block sampling: jointly updating multiple dimensions of a target distribution simultaneously (Roberts and Sahu, 1997; Sargent, Hodges, and Carlin, 2000). When one or more dimensions of the posterior distribution are correlated, joint sampling of these dimensions (with any variety of block samplers) can increase sampling performance relative to updating each dimension independently (*e.g.,* Liu, Wong, and Kong, 1994). Despite wide recognition of the usefulness of this basic idea for designing efficient MCMC algorithms, there has been no automated method for choosing blocks to optimize – or at least greatly improve – performance. Here we develop such a method.

Existing theoretical work comparing block samplers to univariate samplers (Mengersen and Tweedie, 1996; Roberts and Tweedie, 1996; Roberts, Gelman, and Gilks, 1997, among others) has provided many insights but falls short of providing a complete assessment of MCMC efficiency for several reasons. First, it uses MCMC convergence rates as the metric for comparison, without consideration of the computational demands of block sampling. Instead, our viewpoint is that any measure of MCMC efficiency must incorporate both the convergence rate and the computational