Bayesian Analysis (2016) **11**, Number 3, pp. 921–927

Comment on Article by Pratola[*,†]

Christopher M. Hans[‡]

1 Overview

Pratola addresses a specific and challenging problem: the construction of Metropolis–Hastings (MH) proposal mechanisms for regression tree models that are both *efficient* and *effective*. *Efficiency* in this context relates to per-iteration computation time, which is desired to be kept to a minimum. *Effectiveness* in this context relates to the mixing of the resulting chain and its ability to avoid becoming trapped in local modes. As is typical when designing Markov chain Monte Carlo (MCMC) algorithms, these desiderata must be balanced against each other. Moves that are computationally efficient often result in slow-mixing chains, while moves that result in fast-mixing chains—if mechanisms for proposing such moves can even be found—are often accompanied by a high computational burden. Balancing these desiderata is quite difficult, both in general and in the particular case of Bayesian regression tree models.

Pratola's approach to improving MCMC efficiency and effectiveness in the Bayesian regression tree model setting draws on two existing and commonly-used MH moves. Pratola generalizes these moves to be more aggressive in exploring the posterior without sacrificing much computational efficiency. The first is a move based on a rotation mechanism (see, e.g., Sleator et al., 1988) used by Gramacy and Lee (2008) in the sampling of Bayesian treed Gaussian process models. The generalized rotate proposal developed by Pratola allows for nontrivial changes to be made to the interior structure of the tree. Critically, the nodes involved in the rotation need not all split on the same variable, as was the case in earlier implementations. The second is a move that changes a cutpoint and/or splitting variable that has been implemented in various ways in the literature (e.g., Chipman et al., 1998; Dennison et al., 1998; Chipman et al., 2002; Wu et al., 2007; Gramacy and Lee, 2008; Chipman et al., 2010). The generalized perturb-within-change-of-variable move allows for flexibility in moving the cutpoint for a given split while using the covariates to inform the proposal distribution. The resulting generalized moves are *computationally local* yet *structurally global* in that they allow the chain to avoid becoming trapped in local modes while restricting computation at any given iteration to localized regions of the tree.

The goal of balancing computational speed with algorithmic effectiveness arises in many computational settings. When thinking about such a balance, I am often reminded of van Dyk and Meng (1997), who describe a search for "a 'free and better lunch,' not a 'better but expensive lunch' in terms of human and computational effort" in the context of designing ECM (Meng and Rubin, 1993) and ECME (Liu and Rubin, 1994)

*Main article DOI: 10.1214/16-BA999.

†This work was supported by the U.S. National Science Foundation under award number DMS-1310294.

‡Department of Statistics, The Ohio State University, Columbus, Ohio, U.S.A., hans@stat.osu.edu

 DOI: 10.1214/16-BA999B