Bayesian Analysis (2016) **11**, Number 3, pp. 913–919

Comment on Article by Pratola*

Robert B. Gramacy†

I'd like to offer my congratulations to Pratola for engaging in timely study on a high-impact topic, namely the efficient exploration of the space of reasonable partition-based representations of the input–output relationships in data. Tree-based partitioning schemes for regression and classification have proliferated in machine learning, spatial statistics, and computer experiments. However, the Bayesian approach has long been limited by expensive Markov chain Monte Carlo (MCMC) and poor mixing therein.

Pratola said that the MCMC mixing "problem [with trees] has been recognized since such models were established ... and little progress has been made." That's true, but why? MCMC is falling out of fashion a bit, so that may be one explanation. Referees routinely ask authors to remove MCMC details from papers, or at best move them to an appendix, which discourages authors from embarking on the kind of very valuable study that Pratola has taken on in this work. But I think the main reason is that trees are a difficult data structure to deal with. The intersection of talented coders (particularly C data structures), and thoughtful experienced Bayesians, is unfortunately quite small. Not many people are qualified for the job.

My aim over the next several pages is to emphasize, primarily through a series of worked-code illustrations, the value of the contribution Pratola has made. Pratola has provided many of his own illustrations within the Bayesian Additive Regression Tree (BART, Chipman et al., 2010) framework, involving sums of trees, whereas mine will complement those by looking at single-tree models. Following that, I will mention a small potential downside, which I think could be addressed although it may involve a substantial undertaking. Finally, I will conclude with some comments on tree priors, a topic which has been similarly overlooked in the almost two decades since the first swarm of Bayesian tree methods arrived on the scene.

1 An illustration

Pratola talked about rotations, extending an idea from my PhD work (Gramacy, 2005). Whereas my version of rotations worked only on adjacent splits on identical input variables, Pratola's are far more general. Here my aim is to illustrate the value of rotations, and for ease of visualization I shall limit myself to a simple 1-dimensional regression problem.

The data generating mechanism is given by the R code below. This data was used to illustrate treed Gaussian processes in the original methods paper (Gramacy and Lee, 2008) and in the software paper (Gramacy, 2007) for the `tgp` package in R. It is part sinusoid and part linear; a visual will be provided shortly.

*Main article DOI: 10.1214/16-BA999.
†Department of Statistics, Virginia Tech, Blacksburg, VA, rbg@vt.edu

 DOI: 10.1214/16-BA999A

```
R> X <- seq(0,20,length=200)
R> Ztrue <- (sin(pi*X/5) + 0.2*cos(4*pi*X/5)) * (X <= 9.6)
R> Ztrue[X>9.6] <- -1 + X[X>9.6]/10
R> Z <- Ztrue + rnorm(length(Ztrue), sd=0.1)
```

The code below uses a routine from the `tgp` package to fit a so-called "Bayesian CART" surface—the Bayesian analog of the Classification and Regression Tree method of Breiman et al. (1984), which fits piece-wise constant surfaces to the data.

```
R> library(tgp)
R> orig <- bcart(X=X, Z=Z, BTE=c(10000, 1010000, 2), minpart=4, verb=0)
R> orig$gpcs

##         grow     prune    change      swap
## 1 0.01721804 0.0169296 0.3617887 0.7117769
```

The output shown (`out$gpcs`) provides the acceptance rate(s) of tree MCMC moves. The *rotate* move is a special case of *swap*, so the rate quoted above for *swap* combines both *swap* and *rotate* moves. I would consider this mixing to overall be good, perhaps even very good for *change* and *swap*. The mixing indicated is merely "acceptable" for the dimension-changing proposals *grow* and *prune*. Figure 1 shows the data and the resulting predictive surface. Obviously this data wasn't tailor-made for piecewise constant models, but nevertheless the fit is pretty good. You can see the MCMC smoothing over possible splitting locations.

At Pratola pointed out, the trouble comes (in part) when there are confounders in the inputs. A simple way to illustrate that is with a "perfect" confounder: a duplicate x coordinate. The R code below shows what I mean.

```
R> confound <- bcart(X=cbind(X,X), Z=Z, BTE=c(10000, 1010000, 2),
+     minpart=4, verb=0)
R> gpcs <- rbind(orig=orig$gpcs, confound=confound$gpcs)
R> gpcs

##                grow     prune    change      swap
## orig     0.01721804 0.0169296 0.3617887 0.7117769
## confound 0.01221110 0.0118770 0.3630451 0.5479598
```

The tree shouldn't care whether it is partitioning on x_1 or x_2 because they are the same. But since the `tgp` software can't perform rotations on different input variables, only identical adjacent ones, MCMC mixing suffers. Observe the decrease in the rate of accepted *swap*s. There should be no direct effect on *grow* and *prune* rates. However, fewer accepted *rotate*s limits the scope for pruning and re-growth, a problem which is easily exacerbated in higher dimension.

Several remedies have been proposed to improve exploration of the Bayesian tree posterior. The simplest—multiple restarts of the MCMC chain—goes back to the origi-

```
plot(X,Z, cex=0.25)
lines(X, orig$Zp.mean, lwd=2)
lines(X, orig$Zp.q1, lwd=2, lty=2, col=1)
lines(X, orig$Zp.q2, lwd=2, lty=2, col=1)
```

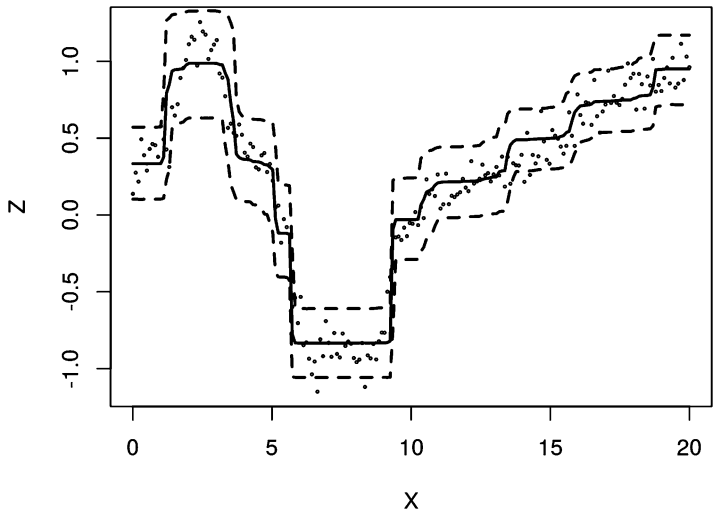

Figure 1: Bayesian CART fit to the sinusoidal data.

nal Bayesian CART paper (Chipman et al., 1998). That feature is facilitated in the `tgp` package via the optional `R=` argument.

```
restart <- bcart(X=cbind(X,X), Z=Z, minpart=4, verb=0,
    BTE=c(10000, 20000, 2), R=100)
gpcs <- rbind(gpcs, restart=restart$gpcs)
gpcs
```

```
##                grow      prune    change      swap
## orig     0.01721804 0.01692960 0.3617887 0.7117769
## confound 0.01221110 0.01187700 0.3630451 0.5479598
## restart  0.02807488 0.01898019 0.3919503 0.5384241
```

Observe that the *swap* rate isn't much affected, but *grow* and *prune* have improved substantially. In fact, they are better than in the original (non-confounded) run. Restarts are a "clunky" way of improving mixing in MCMC, but they do have the implementation advantage of simplicity, and the computational advantage of trivial parallelization. A more involved solution, that has been shown to work well with tree MCMC, involves simulated/parallel tempering (Richardson and Green, 1997). A variation, called importance tempering (Gramacy et al., 2010), which combines simulated tempering and importance sampling (to save samples from heated chains), is implemented in the `tgp` package. Its usage in the package is described in a follow-on tutorial (Gramacy and

Taddy, 2010). That documentation encourages three restarts, to adjust the temperature ladder via stochastic approximation, and a particular form of hierarchical prior to ensure propriety of the prior at hotter temperatures.

```
R> temper <- bcart(X=cbind(X,X), Z=Z, bprior="b0", R=3,
+     BTE=c(10000, 350000, 2), minpart=4, verb=0, itemps=itemps)
gpcs <- rbind(gpcs, temper=temper$gpcs)
gpcs
```

```
##                grow      prune    change      swap
## orig     0.01721804 0.01692960 0.3617887 0.7117769
## confound 0.01221110 0.01187700 0.3630451 0.5479598
## restart  0.02807488 0.01898019 0.3919503 0.5384241
## temper   0.07863162 0.07734743 0.6114380 0.4311797
```

Observe the much improved mixing compared to any of the three previous runs. Strangely, *swaps* benefit less than the others however. The predictive surfaces obtained from the four methods are so similar to the ones from Figure 1 that I do not duplicate them here, to economize on space. I encourage interested readers to re-run the code for themselves if curious.

To conclude this illustration it is worth pointing out that while the remedies explored above have the potential to offer substantial mixing improvements, they are like a band-aid. By contrast, Pratola's re-designed rotate and perturbation moves are more of a surgical procedure. Additionally they may be combined with restarts and tempering to achieve further improvements still.

2 Potential downside?

Much of the Pratola's setup assumes that one has access to an integrated likelihood for the terminal nodes, following similar assumptions in the original Bayesian tree papers. This is almost always the right thing to do, generically, in Bayesian inference: analytically integrate out as much as possible, leaving as little as possible to Monte Carlo integration via MCMC. In the case of the usual leaf models under conjugate priors, for example the constant and multinomial models (CART), and the linear model (i.e., OLS Chipman et al., 2002), these integrated likelihoods are readily available. It is probably also doable for other members of the exponential family, such as the Poisson, Gamma, log Normal, Beta, etc., although I am not aware of any work along these lines.

However, it is clearly not *always* possible, for example in the contexts outlined above without conjugate priors, or with a generalized linear model at the leaves, or with Gaussian processes (GP) at the leaves (Gramacy and Lee, 2008)—a case near and dear to me. In all three situations, the parameters that describe the local fit at the leaves cannot be (fully) integrated out, which means that values of those parameters need to be stored in the data structure at the terminal node, and they need to be updated by the MCMC. MCMC moves which keep the tree topology fixed or nearly-so (e.g.,

change and *swap*), or which condition on the current tree, are straightforward. Several authors have developed simple, efficient, Metropolis schemes for those updates. See, for example, Gramacy and Lee (2008) for the GP case in particular. Tree-moves, however, must involve joint proposals for change in topography as well as leaf-node parameter values. Dimension-changing moves, like *grow* and *prune* must involve proposals for new parameters (and a reversible way to discard them) in a reversible jump-like fashion (Richardson and Green, 1997). Details have been worked out in the GP case, and they essentially involve borrowing of information locally in nearby leaf or parent nodes.

It is not clear, however, whether similar schemes could be extended to Pratola's new rotate move, which in its current form assumes integrated likelihoods at the leaves. Since, by design, the moves make "big" jumps in tree-space, rather than local ones, it is not clear where new values of leaf parameters could come (or be absorbed) from. Moreover, the new perturbation move could drastically alter the composition of terminal nodes, rendering the parameters stored in those nodes useless, at least from the perspective of posterior support relative to the current, unaltered tree before the proposed modifications. So two questions I have for Pratola are: (1) How would you complete the new rotate description to include leaf parameters; and (2) similarly, how would you complete the same for perturbation?

3 Final thought on priors

The *process prior* of Chipman et al. (1998) reigns in tree complexity by penalizing splits on nodes η based on their depth D_η in the tree $\mathcal{T}$:

$$p_{split}(\eta, \mathcal{T}) = \alpha(1 + D_\eta)^{-\beta}.$$

Usually a uniform prior p_{rule} is placed on splitting variables (i.e., splitting dimension) and on locations for splits along those variables. This induces a prior for the full tree $\mathcal{T}$ via the probability that internal nodes $\mathcal{I}_\mathcal{T}$ split and leaves $\mathcal{L}_\mathcal{T}$ do not:

$$\pi(\mathcal{T}) \propto \prod_{\eta \in \mathcal{I}_\mathcal{T}} p_{\text{split}}(\mathcal{T}, \eta) \prod_{\eta \in \mathcal{L}_\mathcal{T}} [1 - p_{\text{split}}(\mathcal{T}, \eta)].$$

In what follows I call this the CGM prior.

In nearly every Bayesian tree paper that I know of, since 1998, this prior has been adopted, or has been used as the basis for a slightly more elaborate process. Why is this prior so popular? Is it any good? Of course, those who know the literature know that there are two notable exceptions to this prior's ubiquity. One is the so-called "pinball" prior of Wu et al. (2007), and the other came from a peer paper (Denison et al., 1998) published in the same year as CGM. The pinball prior has some attractive features, but it is somewhat more complicated to calculate. The latter paper is interesting because it offers something simpler than the CGM prior, in that only the number of splits (before a terminal node) is penalized, with the length of that chain following a truncated Poisson. Lets call that the DMS prior. Denison et al. felt compelled to defend the DMS prior, and thereby their entire Bayesian tree modeling philosophy, relative to the CGM one.

They said:

> The Chipman et al. (1998) approach concentrates more on the prior specification ... to encourage trees to grow to specific topologies ... Although this ... may be beneficial in a few examples, we rarely know the type of tree structure we expect, so we prefer to place no constraint on the structure and let the data speak for themselves.

Apparently, they feel that their Poisson approach is "more uniform" than CGM, or at least gives the data more opportunity to shine. I wonder if that is really the case, although clearly the calculation involves a less cumbersome procedure. They also claim that the DMS's uniformity is better, in the sense that it ought to work better in a generic sense, i.e., for most examples where one would have little *a priori* information about tree structure. My hunch is that this is probably not true, but I the only evidence I have is anecdotal. Nearly twenty years after these papers were published—in 1998, with earlier versions appearing several years previous—we know now the full value of regularization in hard regression and classification problems. We opt for stronger priors, offering more regularization, not less. The CGM prior reins in complexity by controlling tree topography in a more aggressive way than the DMS prior does. You could criticize that it does so arbitrarily, but the jury is out until someone does a proper comparison. The fact that the CGM, not the DMS, has been chosen in several recent papers—on BART, on `tgp`, as part of a dynamic tree prior (Taddy et al., 2011), and an online tree prior (Anagnostopoulos and Gramacy, 2013)—suggests that CGM has emerged as the champion. But nobody has offered a thorough comparison study.

So to conclude, I shall reiterate that Pratola has, in my opinion, sealed the deal on MCMC moves in Bayesian tree models, at least for the next twenty years. I look forward to the R package implementing the new tree moves, because the code behind them isn't the sort of thing that a keen student can recreate for themselves over along weekend, or even over a semester! However, before another twenty years goes by, I'd like to see some re-thinking of the tree prior. The next paper on Bayesian trees should explore the two-or-three existing alternatives, and probably recommend a new one because none of the existing ones provides an interpretable way to specify elucidated prior beliefs.

References

Anagnostopoulos, C. and Gramacy, R. (2013). "Information-Theoretic Data Discarding for Dynamic Trees on Data Streams." *Entropy*, 15(12): 5510–5535. 918

Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth. MR0726392. 914

Chipman, H., George, E., and McCulloch, R. (1998). "Bayesian CART Model Search (with discussion)." *Journal of the American Statistical Association*, 93: 935–960. MR1631325. doi: http://dx.doi.org/10.2307/2670105. 915, 917, 918

Chipman, H., George, E., and McCulloch, R. (2002). "Bayesian Treed Models." *Machine Learning*, 48: 303–324. 916

Chipman, H., George, E., and McCulloch, R. (2010). "BART: Bayesian Additive Regression Trees." *Annals of Applied Statistics*, 4(1): 266–298. MR2758172. doi: http://dx.doi.org/10.1214/09-AOAS285. 913

Denison, D., Mallick, B., and Smith, A. (1998). "A Bayesian CART Algorithm." *Biometrika*, 85: 363–377. MR1649118. doi: http://dx.doi.org/10.1093/biomet/85.2.363. 917

Gramacy, R. B. (2005). "Bayesian Treed Gaussian Process Models." Ph.D. thesis, University of California, Santa Cruz. MR2708095. 913

Gramacy, R. B. (2007). "tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models." *Journal of Statistical Software*, 19(9): 1–46. http://www.jstatsoft.org/v19/i09/ 913

Gramacy, R. B. and Lee, H. K. H. (2008). "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling." *Journal of the American Statistical Association*, 103: 1119–1130. MR2528830. doi: http://dx.doi.org/10.1198/016214508000000689. 913, 916, 917

Gramacy, R. B., Samworth, R. J., and King, R. (2010). "Importance Tempering." *Statistics and Computing*, 20(1). MR2578072. doi: http://dx.doi.org/10.1007/s11222-008-9108-5. 915

Gramacy, R. B. and Taddy, M. (2010). "Categorical Inputs, Sensitivity Analysis, Optimization and Importance Tempering with tgp Version 2, an R Package for Treed Gaussian Process Models." *Journal of Statistical Software*, 33(6): 1–48. http://www.jstatsoft.org/v33/i06/ MR2578072. doi: http://dx.doi.org/10.1007/s11222-008-9108-5. 915

Richardson, S. and Green, P. J. (1997). "On Bayesian Analysis of Mixtures with an Unknown Number of Components." *Journal of the Royal Statistical Society, Series B, Methodological*, 59: 731–758. MR1483213. doi: http://dx.doi.org/10.1111/1467-9868.00095. 915, 917

Taddy, M., Gramacy, R., and Polson, N. (2011). "Dynamic Trees for Learning and Design." *Journal of the American Statistical Association*, 106(493): 109–123. MR2816706. doi: http://dx.doi.org/10.1198/jasa.2011.ap09769. 918

Wu, Y., Thelmeland, H., and West, M. (2007). "Bayesian CART: Prior Structure and MCMC Computations." *Journal of Computation and Graphical Statistics*, 16: 44–66. MR2345747. doi: http://dx.doi.org/10.1198/106186007X180426. 917