Streaming data analysis with dynamic regression trees

Simon Wilson Michael Ferreira

School of Computer Science and Statistics
Trinity College Dublin, Ireland

Context

- Dynamic regression models can be used for active learning and concept drift applications;
- This work explores regression tree models in this context;
- Emphasis is on use for streaming data applications;
- Bases of work are the original work on BART (Chipman et al., 2010)[1] with extension to dynamic case (Taddy et al., 2010) [2].

[1] Chipman, H.A., George, E.I. and McCulloch, R.E., 2010. BART: Bayesian additive regression trees, *Ann. Appl. Statist.*, **4**, 1: 266–298.

[2] Taddy, M., Gramacy, R.B. and Polson, N., 2010. Dynamic Trees for Learning and Design, *J. Amer. Stat. Assoc*, **106**: 109–123.

Idea 1: Dynamic regression models

- Time series $y_t \in \mathbb{R}^n$ depends on explanatory variables $x_t \in \mathbb{R}^m$:

$$y_t = g(x_t; z_t, \theta).$$

- y_t also depends on a latent Markov process $z_t \in \mathbb{R}^k$ defined by $p(z_0 \mid \theta)$, $p(z_t \mid z_{t-1}, \theta)$ and fixed parameters θ;
- Relationship between y_t and x_t *changes with time* through the latent process;
- Complete model to time t looks like:

$$\begin{aligned} &p(y_{1:t}, z_{0:t}, \theta \mid x_{1:t}) \\ &\quad = \; p(z_0 \mid \theta) \left(\prod_{i=1}^{t} p(y_i \mid x_i, z_i, \theta)\, p(z_i \mid z_{i-1}, \theta) \right) p(\theta). \end{aligned}$$

Idea 1: the Kalman filter

- In many respects the simplest example of a dynamic state-space model:

$$\begin{aligned} z_0 &\sim N(\mu_0, W_0); \\ z_{t+1} \mid u_t, z_t &\sim N(F_t z_t + G_t u_t, W_t); \\ y_t \mid z_t &\sim N(H_t z_t, V_t), \end{aligned}$$

  where u_t are additional fixed and known covariates.

- For fixed and known

$$\theta = (\mu_0, F_{0:t+1}, G_{1:t+1}, H_{1:t+1}, W_{0:t+1}, V_{1:t+1}),$$

  closed form expressions for:

  $p(z_t \mid u_{1:t}, y_{1:t}, \theta)$ (*smoothing*);
  $p(z_{t+1} \mid u_{1:t+1}, y_{1:t}, \theta)$ and $p(y_{t+1} \mid u_{1:t+1}, y_{1:t}, \theta)$ (*prediction*).

Idea 1: Kalman filter updating equations

In case you've forgotten!

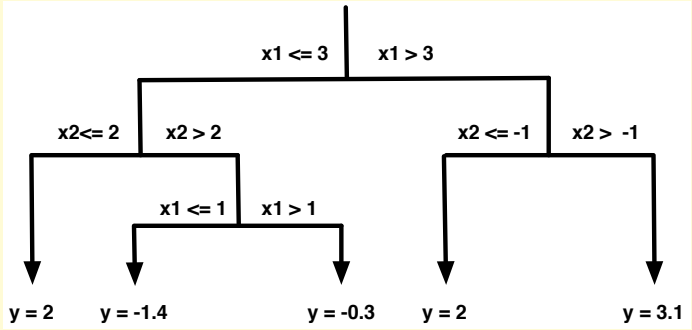
$$\begin{aligned} z_t \mid u_{1:t}, y_{1:t}, \theta &\sim N(\hat{\mu}_t, \hat{\Sigma}_t); \\ z_{t+1} \mid u_{t+1}, u_{1:t}, y_{1:t}, \theta &\sim N(F_{t+1}\hat{\mu}_t + G_{t+1}u_{t+1}, R_{t+1}); \\ y_{t+1} \mid u_{1:t+1}, y_{1:t}, \theta &\sim N(H_{t+1}(F_{t+1}\hat{\mu}_t + G_{t+1}u_{t+1}), \\ &\qquad H_{t+1}^T R_{t+1} H_{t+1} + V_{t+1}), \end{aligned}$$

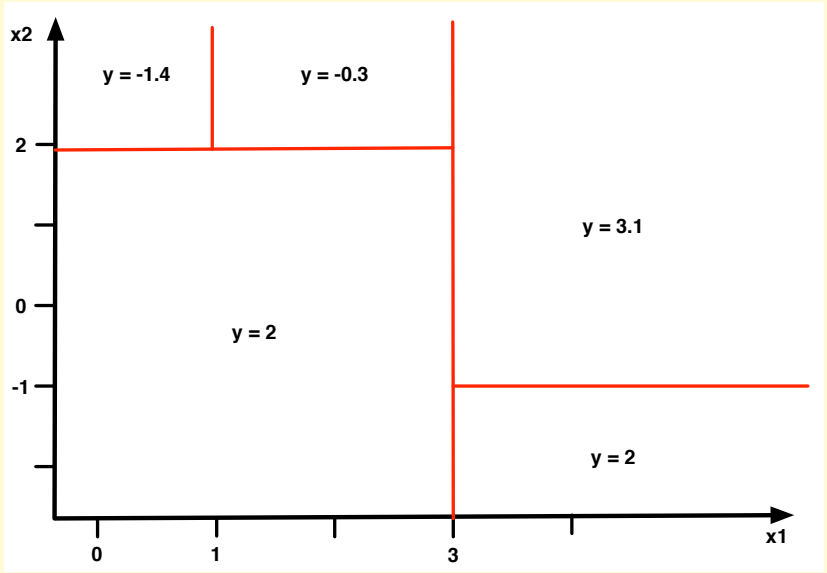
where $R_{t+1} = F_{t+1}^T \hat{\Sigma}_t F_{t+1} + W_{t+1}$, and $\hat{\mu}_t$ and $\hat{\Sigma}_t$ are defined recursively:

$$\begin{aligned} \hat{\mu}_0 &= \mu_0 \text{ and } \hat{\Sigma}_0 = W_0; \\ \hat{\mu}_{t+1} &= F_{t+1}\hat{\mu}_t + R_{t+1}^T H_{t+1}^T (H_{t+1} R_{t+1} H_{t+1}^T + V_{t+1})^{-1} H_{t+1} R_{t+1} \\ &\qquad \times (y_{t+1} - F_{t+1} H_{t+1} \hat{\mu}_t - G_{t+1} u_{t+1}); \\ \hat{\Sigma}_{t+1} &= R_{t+1} - R_{t+1}^T H_{t+1}^T (H_{t+1} R_{t+1} H_{t+1}^T + V_{t+1})^{-1} H_{t+1} R_{t+1}. \end{aligned}$$

Idea 2: Regression trees

- A regression tree $\mathcal{T}$ partitions the space of covariates $\mathcal{X} \subseteq \mathbb{R}^m$ by component-wise splitting rules into (hyper)-rectangles;
- For each partition, the observable $y \in \mathbb{R}^n$ has a fixed value (or more generally a fixed distribution);
- $\mathcal{T}$ parameterized by: splitting component and split threshold at each branch, leaf values (or leaf distribution parameters).





Bringing the 2 ideas together: dynamic Bayesian regression trees

- Goal: produce a reasonably flexible dynamic regression model that still has some tractability;
- Basic idea:
  - y_t follows a regression tree model with explanatory variables x_t and leaf node distributions defined by z_t;
  - z_t is a Markov process so the leaf node distributions evolve in time (so dynamic regression);
  - θ is any other fixed model parameter;
  - Tree structure fixed for now.

Dynamic BART

- Let a tree $\mathcal{T}$ have K terminal nodes;
- Let $\eta(x_t, \mathcal{T}) \in \{1, \ldots, K\}$ be the node index of x_t in the partition defined by $\mathcal{T}$.
- Let $z_t = (z_{t1}, \ldots, z_{tK})$ be partitioned into parameter(s) associated with the distribution of y_t at each node;
- Let θ be any other fixed parameters in the tree;
- Then our dynamic BART model is:

$$\begin{aligned} y_t \,|\, x_t, z_t, \theta, \mathcal{T} &\sim p(y_t \,|\, z_{t,\eta(x_t,\mathcal{T})}, \theta) \\ z_t \,|\, z_{t-1}, \theta &\sim p(z_t \,|\, z_{t-1}, \theta) \end{aligned}$$

- There are priors on z_0, $\mathcal{T}$ and θ.

Example with nice tractability

- Gaussian tree:

$$\begin{aligned} y_t \,|\, x_t, z_t, \theta, \mathcal{T} &\sim N(z_{t,\eta(x_t,\mathcal{T})}, V_{\eta(x_t,\mathcal{T})}), \\ z_{t+1,k} \,|\, z_{tk} &\sim N(F_k z_{tk}, W_k), \; k = 1, \ldots, K, \end{aligned}$$

  with $z_{0k} \sim N(\mu_{0k}, W_{0k})$.
- So y_t is Gaussian and its mean at each node in the tree evolves as an *independent* Gaussian process.
- Nice property:
  - Each node k is an independent Kalman filter
  - ... but we only observe y_t at node k when $\eta(x_t, \mathcal{T}) = k$;

Intermittent Kalman filter

- This is an *intermittent Kalman filter* — closed form expressions still for posteriors like $p(z_t \mid x_{1:t}, y_{1:t}, \theta)$ etc.[3]:
- If you observe y_{t+1} then update $\hat{\mu}_{t+1}$ and $\hat{\Sigma}_{t+1}$ from $\hat{\mu}_t$ and $\hat{\Sigma}_t$ in the usual way;
- If you do not observe y_{t+1} then $\hat{\mu}_{t+1} = \hat{\mu}_t$ and $\hat{\Sigma}_{t+1} = R_{t+1}$.

[3] Sinopoli et al. (2004), Kalman filtering with intermittent observations, *IEEE Transactions on Automatic Control* **49**, 9: 1453–1464.

Inference on the tree structure

- Taddy et al. (2010) proposed a similar model that incorporated learning about the tree structure;
- They had a prior on the tree structure (see also Chipman et al., 2010) and allowed it to evolve by merging/splitting/deleting splits;
- For any tree node:

$$p(\eta \text{ is a split}) \propto \alpha(1+d_\eta)^{-\beta},\ \alpha \in (0,1),\ \beta \geq 0,$$

where d_η is the depth of the node in the tree, then

$$p(\mathcal{T}) \propto \prod_{\substack{\eta \in \mathcal{T} \\ \text{is a split}}} p(\eta \text{ is a split}) \prod_{\substack{\eta \in \mathcal{T} \\ \text{is a leaf}}} (1 - p(\eta \text{ is a split})).$$

Inference on the tree structure

$$
\begin{aligned}
&p(\mathcal{T}, z_{0:t} \,|\, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t}) \\
&\qquad = \; p(\mathcal{T} \,|\, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t})\; p(z_{0:t} \,|\, \mathcal{T}, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t}) \\
&\qquad = \; p(\mathcal{T} \,|\, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t}) \prod_{k=1}^{K_{\mathcal{T}}} p(z_{tk} \,|\, \mathcal{T}, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t}).
\end{aligned}
$$

- The product is of Gaussian terms (conditioned on $\mathcal{T}$);
- Remains to compute $p(\mathcal{T} \,|\, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t})$;
- A recursive formula has been developed that allows $O(1)$ updating at each new observation.
- Ensemble inference used:
  - Fixed ensemble, compute $p(\mathcal{T} \,|\, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t})$ for each tree, use model averaging for prediction;
  - Availability of $p(\mathcal{T} \,|\, \theta_{\mathcal{T}}, x_{1:t}, y_{1:t})$ allows MCMC exploration of space of trees.

Example

- Compared to the Kalman Filter;
- Time-series data set simulated from the Mackey-Glass non-linear time series.
- Ensemble of 50 trees;
- Known parameters:

$$H = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad F = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad W = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad V = 0.03;$$

for $y \in \mathbb{R}, \quad z \in \mathbb{R}^2$.

Example

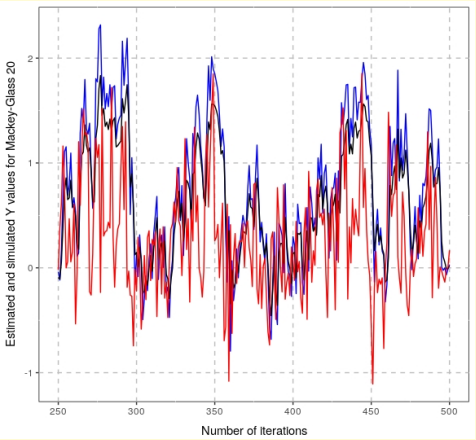


Figure: One-step ahead predictions with BDRT and the Kalman Filter

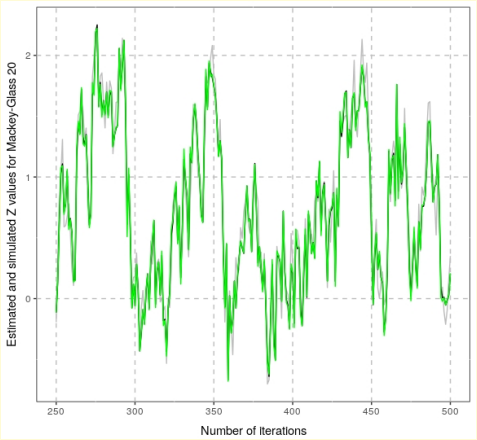


Figure: Latent state predictions with BDRT and the Kalman Filter

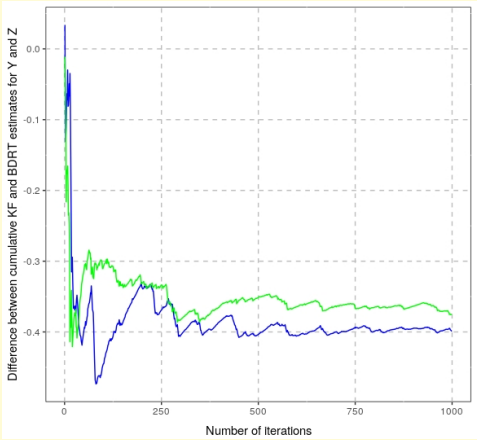


Figure: Difference in RMSE between BDRT and the Kalman Filter

Example

Comparing different MCMC methods:

1. "MH" is the Chipman et al. method using grow, prune, swap and change;
2. "BST" uses the same moves but has simulated tempering with 10 levels of heating. Prior specified using Geyer (1995)[4].
3. "MST" uses different moves: multigrow, multiprune, multichange, shift, and swap. The upper bound of growing, changing and pruning is temperature dependent.

[4] Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. J. Americ Statist. Assoc. **103**, 1119–1130.

Example

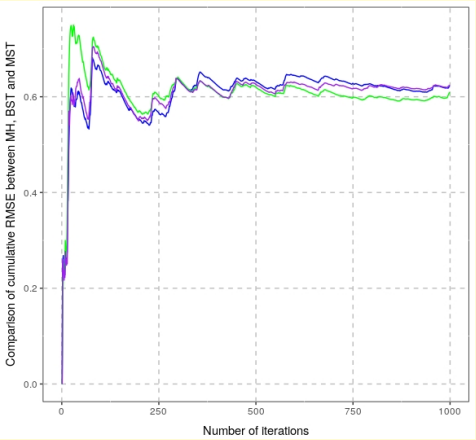


Figure: Comparing RMSE between the 3 different MCMC approaches.

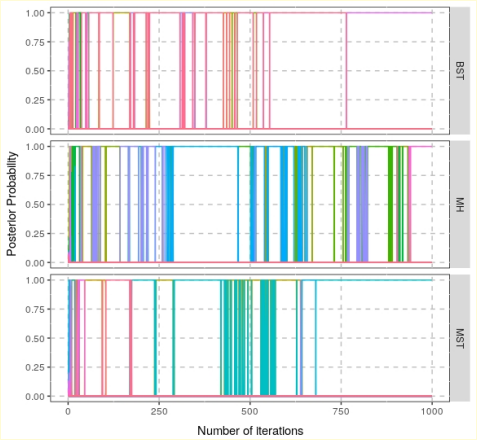


Figure: The probability of the trees alternate between zero and one.

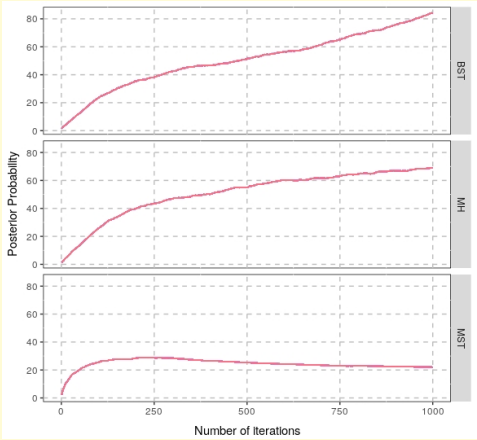


Figure: Average tree size as the algorithm progresses

Example

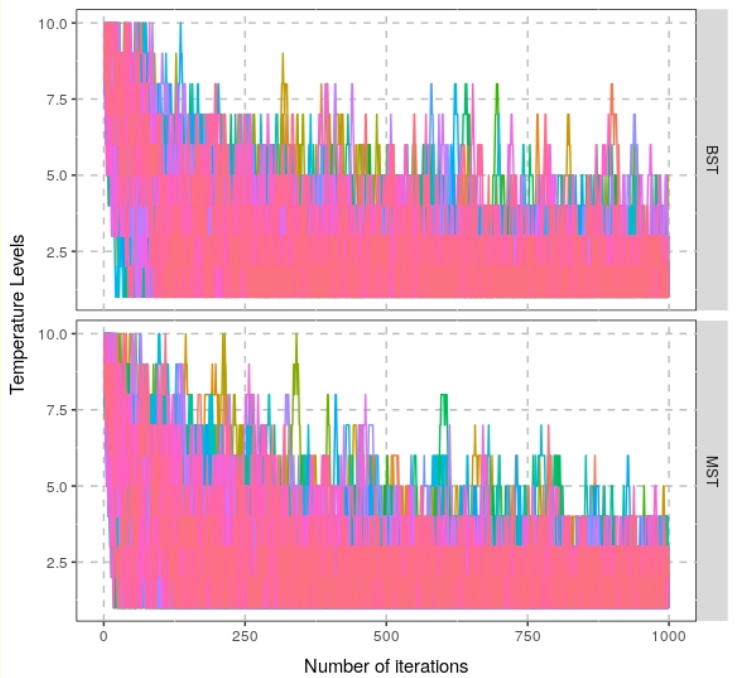


Figure: Temperature traversal of the trees started at the highest temperature.

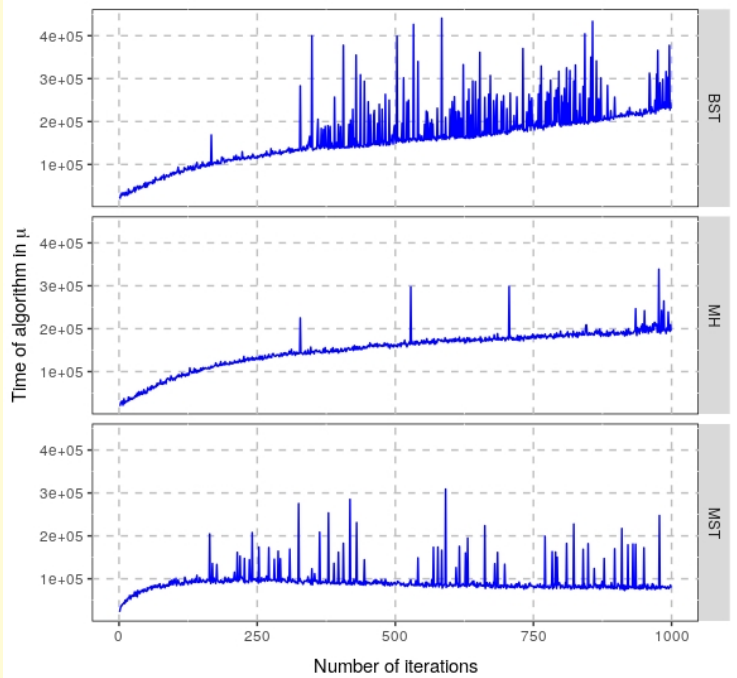


Figure: Time comparisons between different MCMC methods.

To conclude: a note on streaming

- Exchangeability of responses based on conditional data allows us to to develop a window-like streaming algorithm;
- If the data are arriving faster than they can be processed then we can randomly select inputs to process;
- We'll still be in the intermittent Kalman filter model;
- Ability to process depends on tree size, rate of data arrival, speed of algorithm, choice of model type (state only estimation, dual estimation, parameter learning, variable or model selection).

Thank you