Top-down particle filtering for Bayesian decision trees

Balaji Lakshminarayanan[1], Daniel M. Roy[2] and Yee Whye Teh[3]

1. Gatsby Unit, UCL, 2. University of Cambridge and 3. University of Oxford

Outline

Outline

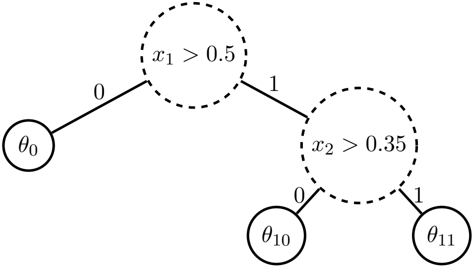
Introduction

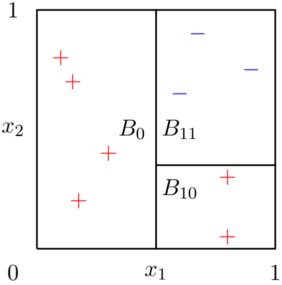
- **Input**: attributes $X = \{x_i\}_{i=1}^N$, labels $Y = \{y_i\}_{i=1}^N$ (i.i.d)
- $y_i \in \{1, \ldots, K\}$ (classification) or $y_i \in \mathbb{R}$ (regression)
- **Goal**: Model $p(y|x)$

Introduction

- **Input**: attributes $X = \{x_i\}_{i=1}^N$, labels $Y = \{y_i\}_{i=1}^N$ (i.i.d)
- $y_i \in \{1, \ldots, K\}$ (classification) or $y_i \in \mathbb{R}$ (regression)
- **Goal**: Model $p(y|x)$
- Assume $p(y|x)$ is specified by decision tree $\mathcal{T}$
- **Bayesian decision trees**:
  - Posterior: $p(\mathcal{T}|Y,X) \propto \underbrace{p(Y|\mathcal{T},X)}_{likelihood}\underbrace{p(\mathcal{T}|X)}_{prior}$
  - Prediction: $p(y_*|x_*) = \sum_{\mathcal{T}} p(\mathcal{T}|Y,X)p(y_*|x_*,\mathcal{T})$

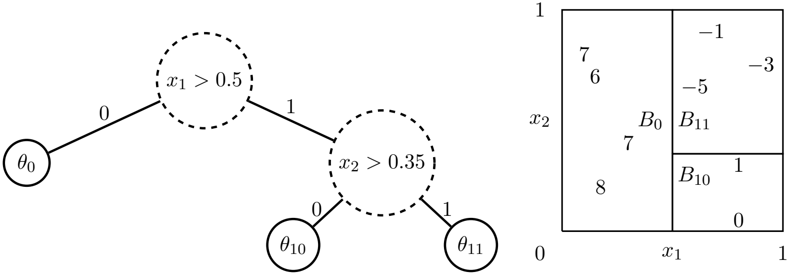
Example: Classification tree





$\boldsymbol{\theta}$: Multinomial parameters at leaf nodes

Example: Regression tree



$\boldsymbol{\theta}$: Gaussian parameters at leaf nodes

Motivation

- Classic non-Bayesian induction algorithms (e.g. CART) learn a single tree in a top-down manner using greedy heuristics (post-pruning and/or bagging necessary)
- MCMC for Bayesian decision trees: [Chipman et al., 1998]: local Monte Carlo modifications to the tree structure (less prone to over fitting but slow to mix)
- **Our contribution:** Sequential Monte Carlo (SMC) algorithm that approximates the posterior, in a top-down manner
- **Take home message**: SMC provides better computation vs predictive performance tradeoff than MCMC

Bayesian decision trees: likelihood

$$p(\mathcal{T}|Y, X) \propto \underbrace{p(Y|\mathcal{T}, X)}_{\textit{likelihood}} \underbrace{p(\mathcal{T}|X)}_{\textit{prior}}$$

Likelihood

- Assume x_n falls in the j^{th} leaf node of $\mathcal{T}$
- Likelihood for n^{th} data point: $p(y_n \mid x_n, \mathcal{T}, \boldsymbol{\theta}) = p(y_n|\theta_j, x_n)$

$$p(Y \mid \mathcal{T}, X, \boldsymbol{\Theta}) = \prod_n p(y_n \mid x_n, \mathcal{T}, \boldsymbol{\theta}) = \prod_{j \in \text{leaves(T)}} \prod_{n \in N(j)} p(y_n|\theta_j)$$

Likelihood

- Assume x_n falls in the j^{th} leaf node of $\mathcal{T}$
- Likelihood for n^{th} data point: $p(y_n \mid x_n, \mathcal{T}, \boldsymbol{\theta}) = p(y_n|\theta_j, x_n)$

$$p(Y \mid \mathcal{T}, X, \boldsymbol{\Theta}) = \prod_n p(y_n \mid x_n, \mathcal{T}, \boldsymbol{\theta}) = \prod_{j \in \text{leaves(T)}} \prod_{n \in N(j)} p(y_n|\theta_j)$$

- Better: integrate out θ_j, use marginal likelihood

$$p(Y \mid \mathcal{T}, X) = \prod_{j \in \text{leaves(T)}} \int_{\theta_j} \prod_{n \in N(j)} p(y_n|\theta_j) p(\theta_j) d\theta_j$$

- Classification: Dirichlet - Multinomial
- Regression: Normal - Normal Inverse Gamma

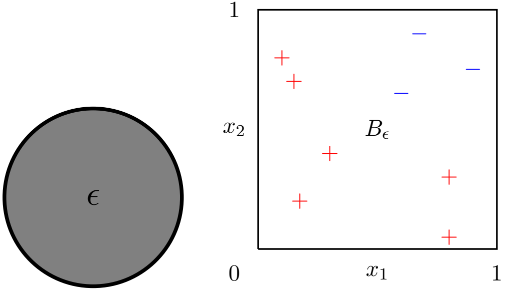
Outline

Bayesian decision trees: prior

$$p(\mathcal{T}|Y,X) \propto \underbrace{p(Y|\mathcal{T},X)}_{likelihood}\,\underbrace{p(\mathcal{T}|X)}_{prior}$$

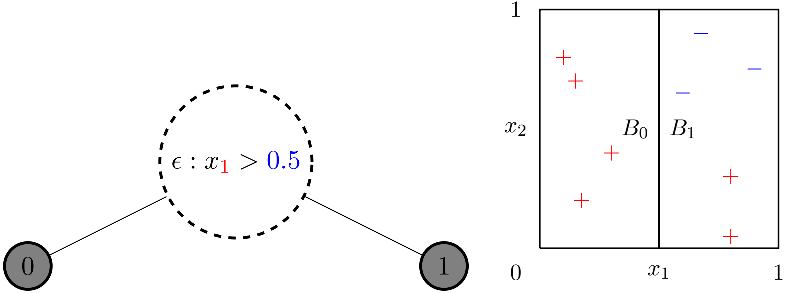
Partial trees

0. Start with empty tree.



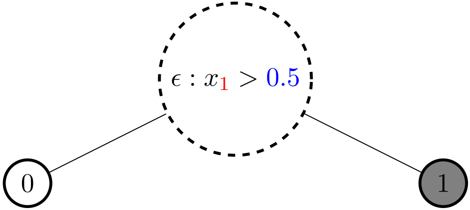
Partial trees

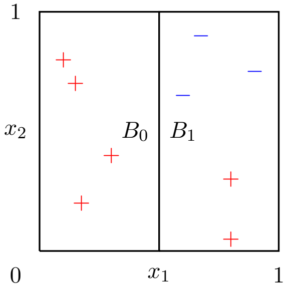
1. Choose to split root node with feature 1 and threshold 0.5.



Partial trees

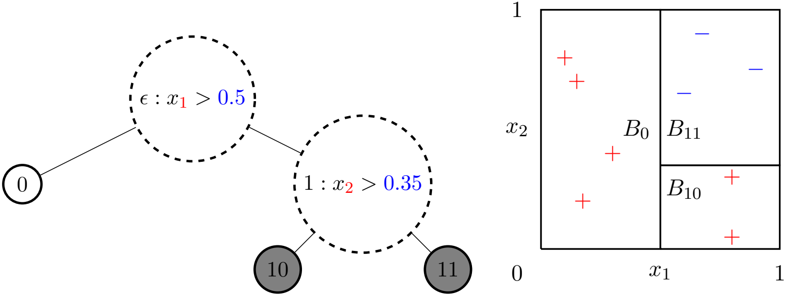
2. Choose to not split node 0.





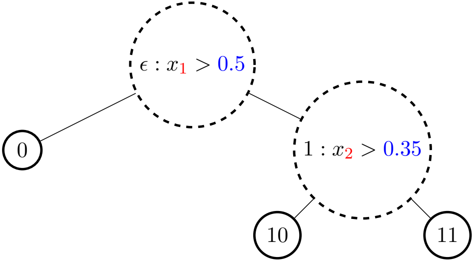
Partial trees

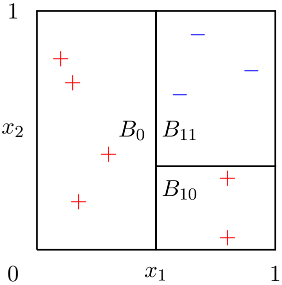
3. Choose to split node 1 with with feature 2 and threshold 0.35.



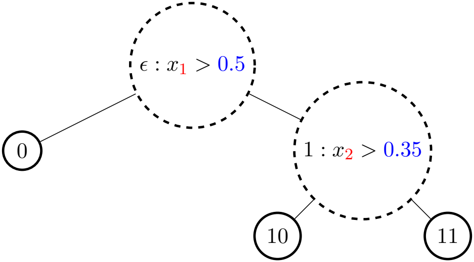
Partial trees

4. Choose to not split node 10.
5. Choose to not split node 11.





Sequence of random variables for a tree



1. $\rho_\epsilon = 1, \kappa_\epsilon = 1, \tau_\epsilon = 0.5$
2. $\rho_0 = 0$
3. $\rho_1 = 1, \kappa_1 = 2, \tau_1 = 0.35$
4. $\rho_{10} = 0$
5. $\rho_{11} = 0$

Sequential prior over decision trees

- Probability of split (assuming a valid split exists):

$$p(\text{j split}) = \alpha_s \cdot \Big(1 + \mathit{depth}(j)\Big)^{-\beta_s} \quad \alpha_s \in (0,1),\ \beta_s \in [0,\infty)$$

- κ_j, τ_j sampled uniformly from the range of valid splits

Sequential prior over decision trees

- Probability of split (assuming a valid split exists):

$$p(\text{j split}) = \alpha_s \cdot \Big(1 + depth(j)\Big)^{-\beta_s} \quad \alpha_s \in (0,1),\ \beta_s \in [0, \infty)$$

- κ_j, τ_j sampled uniformly from the range of valid splits
- Prior distribution:

$$p(\mathsf{T}, \kappa, \tau | X) = \prod_{j \in \text{leaves}(\mathsf{T})} p(\text{j not split}) \times \prod_{j \in \text{nonleaves}(\mathsf{T})} p(\text{j split}) p(\kappa_j, \tau_j)$$

Outline

Bayesian decision trees: posterior

$$p(\mathcal{T}|Y,X) \propto \underbrace{p(Y|\mathcal{T},X)}_{\textit{likelihood}}\,\underbrace{p(\mathcal{T}|X)}_{\textit{prior}}$$

SMC algorithm for Bayesian decision trees

- Importance sampler: Draw $\mathcal{T}^{(c)} \sim q(\cdot)$

$$p(Y|X) = \sum_{\mathcal{T}} p(Y, \mathcal{T}|X) \approx \sum_{c=1}^{C} \underbrace{\frac{1}{C}\ \frac{p(\mathcal{T}^{(c)})}{q(\mathcal{T}^{(c)})}\ p(Y|X, \mathcal{T}^{(c)})}_{w^{(c)}}$$

SMC algorithm for Bayesian decision trees

- Importance sampler: Draw $\mathcal{T}^{(c)} \sim q(\cdot)$

$$p(Y|X) = \sum_{\mathcal{T}} p(Y, \mathcal{T}|X) \approx \sum_{c=1}^{C} \underbrace{\frac{1}{C} \, \frac{p(\mathcal{T}^{(c)})}{q(\mathcal{T}^{(c)})} \, p(Y|X, \mathcal{T}^{(c)})}_{w^{(c)}}$$

- Normalize: $\bar{w}^{(c)} = \frac{w^{(c)}}{\sum_{c'} w^{(c')}}$
- Approximate posterior:

$$p(\mathcal{T}|Y, X) \approx \sum_{c} \bar{w}^{(c)} \, \delta(\mathcal{T} = \mathcal{T}^{(c)})$$

SMC algorithm for Bayesian decision trees (contd.)

- Sequential importance sampler (SIS):

$$p(\mathcal{T}_n) = p(\mathcal{T}_0) \prod_{n'=1}^{n} p(\mathcal{T}_{n'}|\mathcal{T}_{n'-1}) \quad q(\mathcal{T}_n) = q_0(\mathcal{T}_0) \prod_{n'=1}^{n} q_{n'}(\mathcal{T}_{n'}|\mathcal{T}_{n'-1})$$

$$p(Y|X,\mathcal{T}_n) = \cancel{p(Y|X,\mathcal{T}_0)} \frac{\cancel{p(Y|X,\mathcal{T}_1)}}{\cancel{p(Y|X,\mathcal{T}_0)}} \cdots \frac{p(Y|X,\mathcal{T}_n)}{\cancel{p(Y|X,\mathcal{T}_{n-1})}}$$

SMC algorithm for Bayesian decision trees (contd.)

- Sequential importance sampler (SIS):

$$p(\mathcal{T}_n) = p(\mathcal{T}_0) \prod_{n'=1}^{n} p(\mathcal{T}_{n'}|\mathcal{T}_{n'-1}) \quad q(\mathcal{T}_n) = q_0(\mathcal{T}_0) \prod_{n'=1}^{n} q_{n'}(\mathcal{T}_{n'}|\mathcal{T}_{n'-1})$$

$$p(Y|X,\mathcal{T}_n) = \cancel{p(Y|X,\mathcal{T}_0)} \frac{\cancel{p(Y|X,\mathcal{T}_1)}}{\cancel{p(Y|X,\mathcal{T}_0)}} \cdots \frac{p(Y|X,\mathcal{T}_n)}{\cancel{p(Y|X,\mathcal{T}_{n-1})}}$$

$$\begin{aligned} w &= \frac{1}{C} \, \frac{p(\mathcal{T}_n)}{q(\mathcal{T}_n)} \, p(Y|X,\mathcal{T}_n) \\ &= w_0 \prod_{n'=1}^{n} \frac{p(\mathcal{T}_{n'} \mid \mathcal{T}_{n'-1})}{q_{n'}(\mathcal{T}_{n'} \mid \mathcal{T}_{n'-1})} \underbrace{\frac{p(Y \mid X, \mathcal{T}_{n'})}{p(Y \mid X, \mathcal{T}_{n'-1})}}_{local\ \ likelihood} \end{aligned}$$

- Sequential Monte Carlo (SMC): SIS + adaptive resampling steps
- Every node is processed just once: no **multi-path** issues

Outline

Experimental setup

- Datasets:
  - *magic-04*: $N = 19K$, $D = 10$, $K = 2$.
  - *pendigits*: $N = 11K$, $D = 16$, $K = 10$.
- 70% - 30% train-test split
- Numbers averaged across 10 different initializations

Outline

SMC design choices

- Proposals
  - *prior* proposal: $q_n(\rho_j, \kappa_j, \tau_j) = p(\rho_j, \kappa_j, \tau_j)$
  - *optimal* proposal:

$$
\begin{aligned}
q_n(\rho_j = \text{stop}) &\propto p(\text{j not split}) p(Y_{N(j)} | X_{N(j)}), \\
q_n(\rho_j = \text{split}, \kappa_j, \tau_j) &\propto p(\text{j split}) p(\kappa_j, \tau_j) \\
&\quad \times \underbrace{p(Y_{N(j0)} | X_{N(j0)})}_{\text{left child}} \underbrace{p(Y_{N(j1)} | X_{N(j1)})}_{\text{right child}}.
\end{aligned}
$$

- Set of nodes considered for expansion at iteration n
  - *node-wise*: next node
  - *layer-wise*: all nodes at depth n
- Multinomial resampling

Effect of SMC design choices

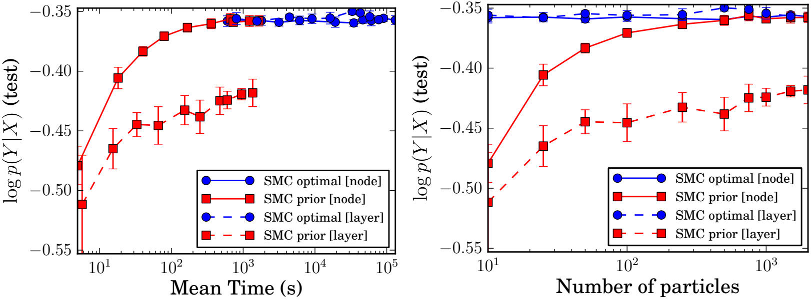


Figure: Results on *magic-04* dataset

Effect of irrelevant features on SMC design choices

madelon: $N = 2.6K$, $D = 500$, $K = 2$
(96% of the features are irrelevant)

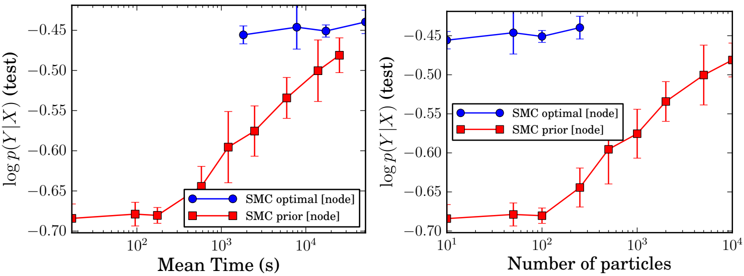


Figure: Results on *madelon* dataset

Outline

Predictive performance vs computation: SMC vs MCMC

- Fix hyper parameters $\alpha = 5, \alpha_s = 0.95, \beta_s = 0.5$
- MCMC [Chipman et al., 1998]: one of the 4 proposals:
  - *grow*
  - *prune*
  - *change*
  - *swap*
- MCMC averages predictions over all previous trees
- Vary number of particles in SMC, number of MCMC iterations and compare runtime vs performance

Predictive performance vs computation: SMC vs MCMC

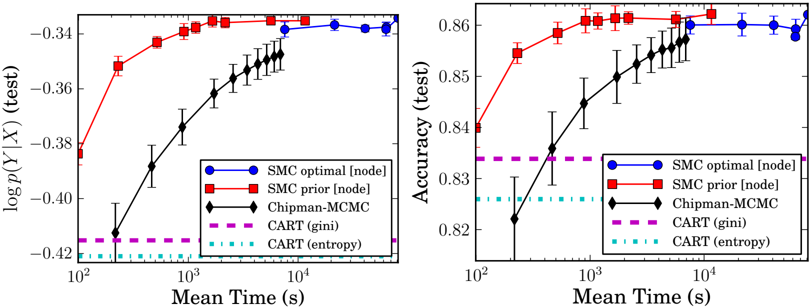


Figure: Results on *magic-04* dataset

Take home message

SMC (*prior*, *node-wise*) is **at least**
an order of magnitude faster than MCMC

Outline

Conclusion

- SMC for fast Bayesian inference for decision trees
  - mimick the top-down generative process of decision trees
  - use ‘local’ likelihoods + resampling steps to guide tree growth
  - For a fixed computational budget, SMC outperforms MCMC

Conclusion

- SMC for fast Bayesian inference for decision trees
  - mimick the top-down generative process of decision trees
  - use ‘local’ likelihoods + resampling steps to guide tree growth
  - For a fixed computational budget, SMC outperforms MCMC
- Future directions
  - Particle-MCMC for Bayesian Additive Regression Trees
  - Mondrian process prior: projective and exchangeable prior for decision trees [Roy and Teh, 2009]

Thank you!

Code available at
http://www.gatsby.ucl.ac.uk/~balaji

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998).
Bayesian CART model search.
J. Am. Stat. Assoc., pages 935–948.

Roy, D. M. and Teh, Y. W. (2009).
The Mondrian process.
In *Adv. Neural Information Proc. Systems*, volume 21, pages 1377–1384.