Random Forests in Theory and in Practice

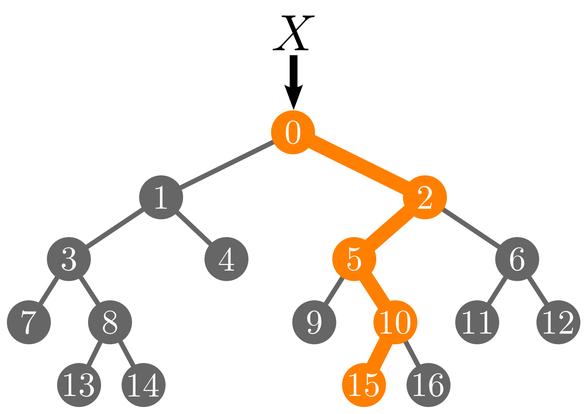
Misha Denil[1] David Matheson[2] Nando de Freitas[1]
[1]University of Oxford [2]University of British Columbia

Motivation

Despite widespread interest and practical use, the theoretical properties of random forests are still not well understood. In this paper we contribute to this understanding in two ways. We present a new theoretically tractable variant of random regression forests and prove that our algorithm is consistent. We also provide an empirical evaluation, comparing our algorithm and other theoretically tractable random forest models to the random forest algorithm used in practice. Our experiments provide insight into the relative importance of different simplifications that theoreticians have made to obtain tractable models for analysis.

Text

Leaf expansion order



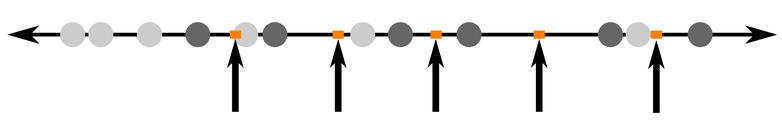
- Breiman: Depth first until minimum leaf size is reached.
- Biau08: Choose leaf uniformly at random.
- Biau12: Breadth first until maximum number of leafs is reached.
- Ours: Depth first until minimum leaf size is reached.

Dimension selection

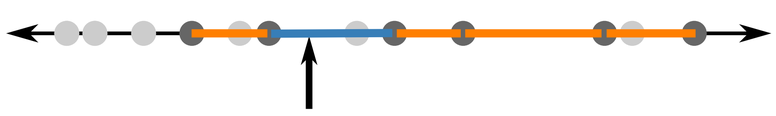
- Breiman: Choose a fixed number of random candidate dimensions without replacement.
- Biau08: Choose a single dimension uniformly at random.
- Biau12: Choose a fixed number of random candidate dimensions with replacement.
- Ours: Choose $\min(1 + \text{Poisson}(\lambda), D)$ candidate dimensions without replacement.

Split point selection

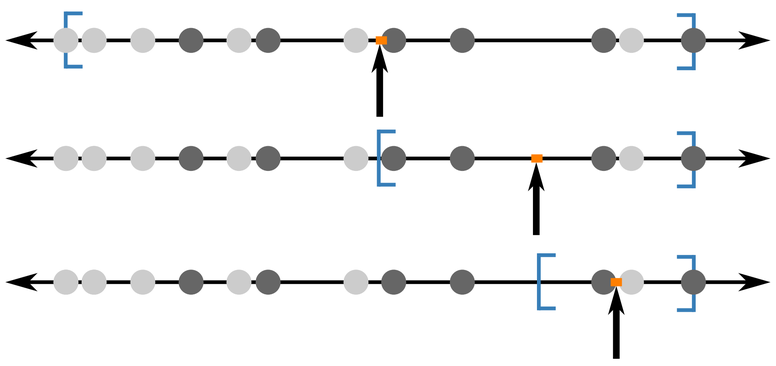
- Breiman: Check the midpoint of every gap and choose the one with the greatest information gain.



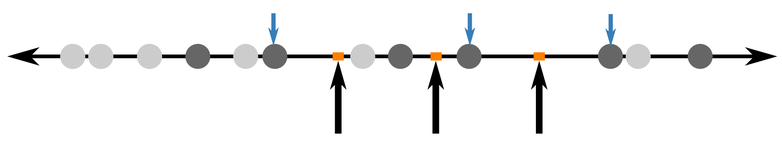
- Biau08: Select a point uniformly at random in a uniformly chosen gap.



- Biau12: Select the midpoint in each cell.



- Ours: Select a few structure points at random and search the midpoint every gap between them for the split that gives the optimal information gain (on estimation points).



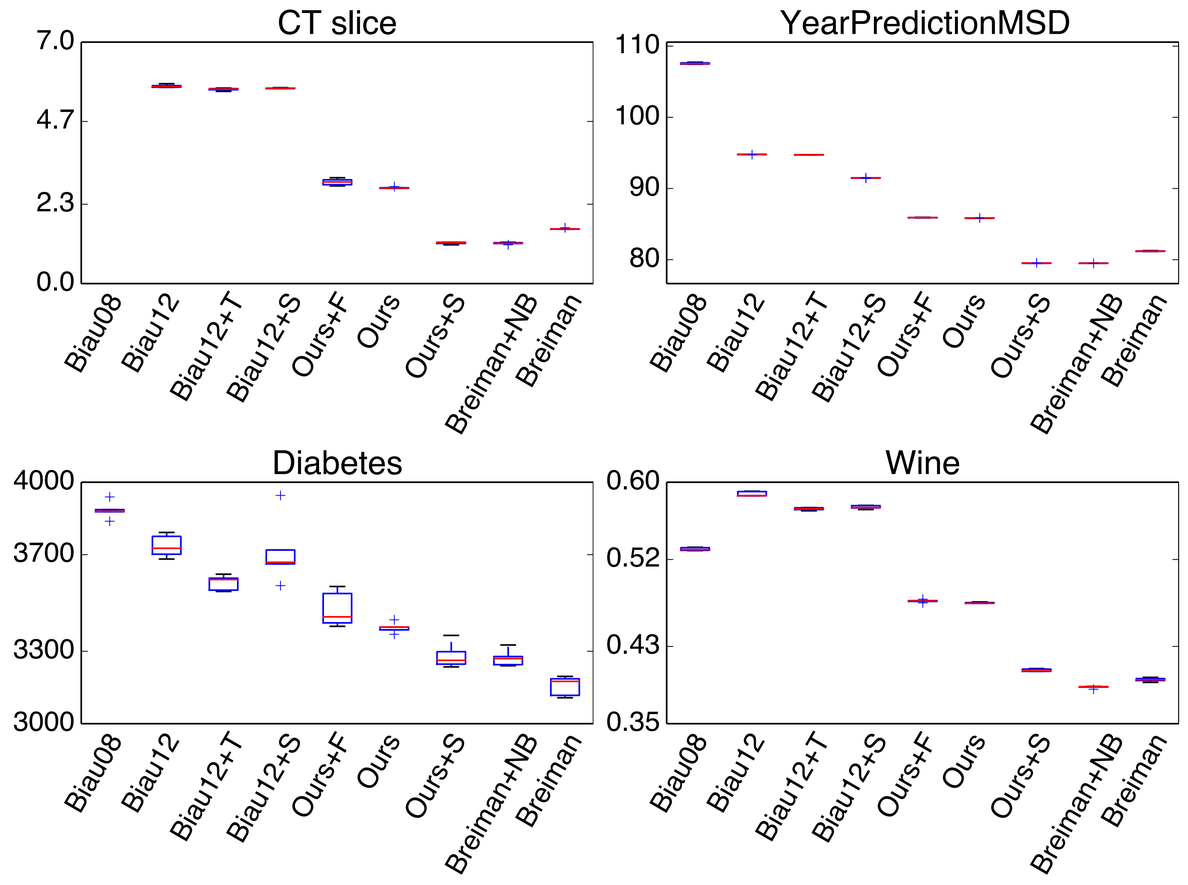
Data Partitioning

- Breiman: No partitioning.
- Biau08: No partitioning.
- Biau12: Structure/estimation partitioning.
- Ours: Structure/estimation partitioning.

Consistency

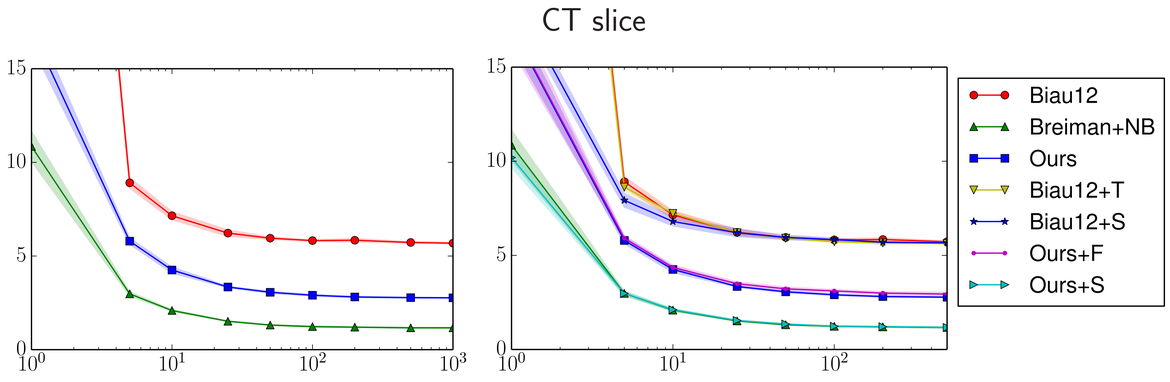
We prove that our random forest algorithm, including the modifications we have made to the dimension and split selection procedures, is consistent. We achieve the closest match to date between tractable and practical algorithms.

Comparison on different data sets



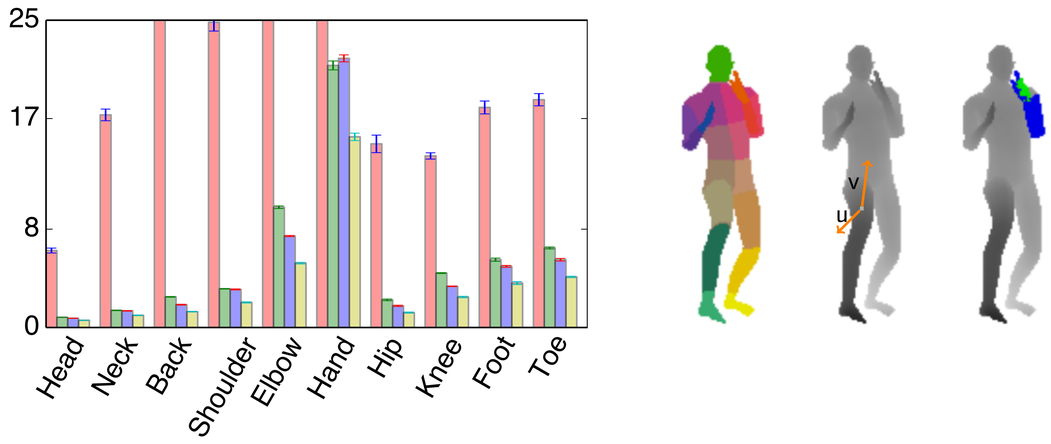
- The y-axis shows MSE.
- +T and +F indicate partitioning at the tree and forest level, respectively.
- +S indicates no partitioning.
- Breiman+NB is Breiman's algorithm with no bootstrapping.

Comparison as a function of forest size



- **Left:** Performance comparison as a function of forest size.
- **Right:** Comparison between dierent methods of data splitting and split point selection on the CT slice dataset.
- In both plots the x-axis is number of trees and the y-axis is MSE.

Kinect pose estimation



- **Bar groups:** Biau08, Baiu12, Ours, Breiman
- **Task:** predict joint location from a (labelled) depth image.
- **Features:** Depth difference at pairs of pixel offsets chosen from a 2d Gaussian.
- For each joint we train a forest on the pixels of the body associated with that joint and predict the relative offset from each pixel to the joint.
- Data generated by sampling random poses from the CMU mocap data set and generating depth images.

Random Forests in Theory and in Practice

Misha Denil[1] David Matheson[2] Nando de Freitas[1]
[1]University of Oxford [2]University of British Columbia

Random Forests in Theory and in Practice

Misha Denil[1] David Matheson[2] Nando de Freitas[1]
[1]University of Oxford [2]University of British Columbia