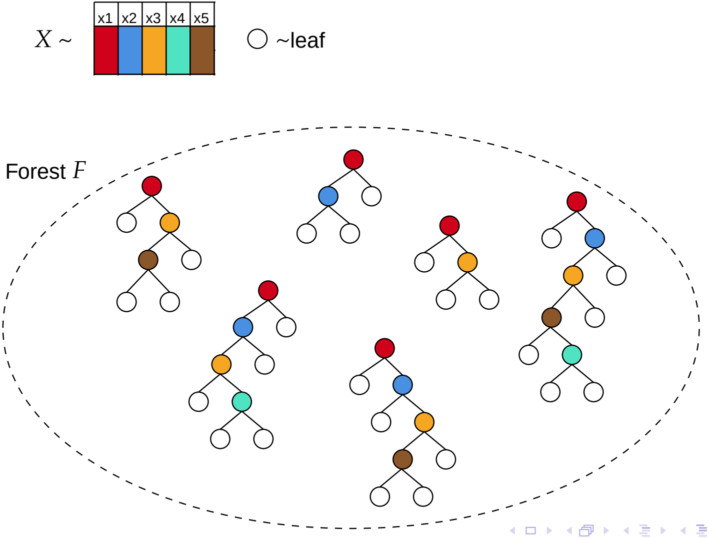
ControlBurn: Feature Selection by Sparse Forests

Brian Liu, Miaolan Xie, and Madeleine Udell

Ensemble Learning

- Given training data $X \in \mathbb{R}^{mxp}$ and response $y \in \mathbb{R}^m$
- Fit a collection of base learners $T_1(x), T_2(x), \dots T_n(x)$ on the training data
- Combine the predictions of the base learners
- **Example:** Regression averaging: $\hat{Y} = \frac{1}{n}\sum_{j=1}^{n} T_j(x)$

Tree Ensembles



Ensemble Post-Processing

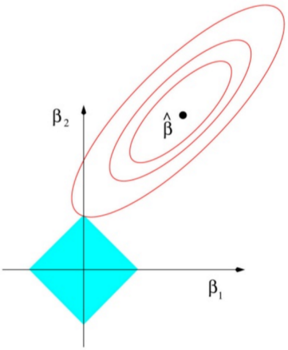
- Reduces ensemble size to:
  - Prevent overfitting
  - Improve interpretability
- Friedman and Popescu (2003): ℓ_1 post processing
- Minimize w.r.t. α

$$\sum_{i=1}^{m} L(y_i, \alpha_0 + \sum_{j=1}^{n} \alpha_j T_j(x_i)) + \lambda \sum_{j=1}^{n} \|\alpha_j\|_1 \qquad (1)$$

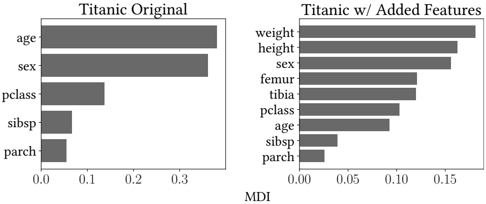
- Loss function $L(y, \hat{y})$:
  - Square loss (Regression): $\|y - \hat{y}\|_2^2$
  - Hinge loss (Classification): $[1 - y\hat{y}]_+$

L1 regularization

- Induces sparsity, coefficients can shrink to zero.
- Example: LASSO regression selects single feature from a group of features



Motivating Example



- Correlation bias: Interpretability ↓

Quiz

If we fit logistic LASSO regression on the Titanic dataset w/ correlated features what is most likely to occur?

A) None of the correlated features will be included in the model.

B) All of the correlated features will be included in the model, with similar coefficients.

C) Only one of the correlated features will be included in the model.

Feature Sparse Ensembles

- **Goal**: Select a subset of learners such that the resulting ensemble does not use all the features
- Important for tree ensembles since they distribute feature importance evenly amongst correlated features

Feature Sparse LASSO for Tree Ensembles

Given: feature matrix $X \in \mathbb{R}^{m \times p}$, response $y \in \mathbb{R}^m$, loss function L

Grow a forest of n trees.

Solve:

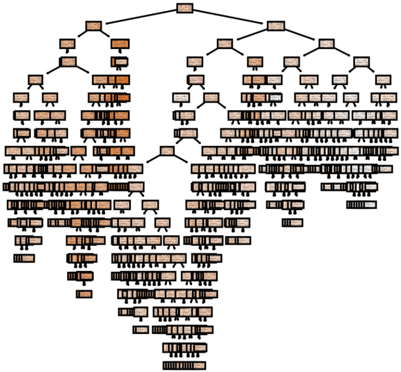
$$\begin{array}{ll} \text{minimize} & \frac{1}{m} L(y, Aw) + \lambda \sum_{i=1}^{n} u_i w_i \\ \text{subject to} & w \geq 0 \end{array} \tag{2}$$

$A \in \mathbb{R}^{m \times n}$: predictions of each tree as columns.

u_i is the number of features used in tree i.

Problem

- What if every tree uses all the features?
- Either all or none of the features will be selected.



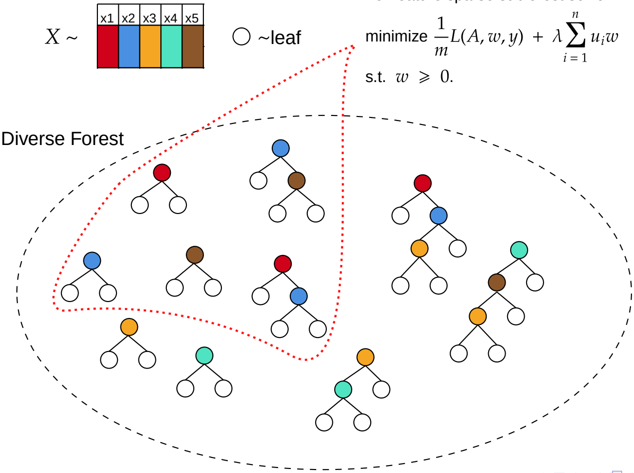
Solution

Grow a **diverse** forest.

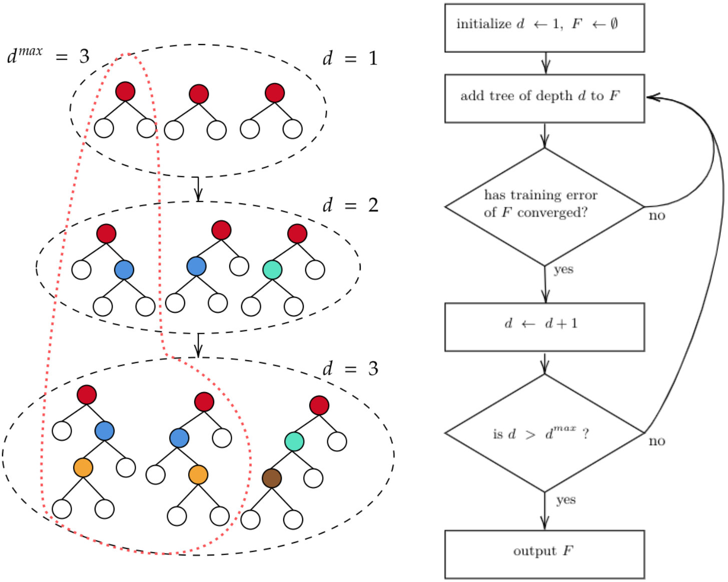
For feature sparse subforest solve:

minimize $\frac{1}{m} L(A, w, y) \; + \; \lambda \sum_{i=1}^{n} u_i w$

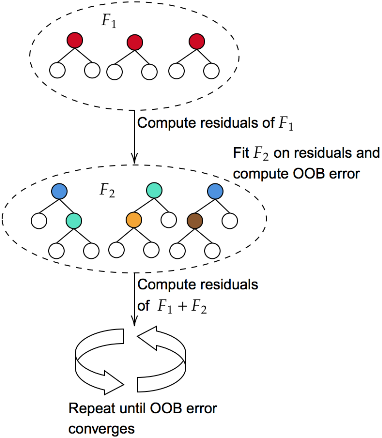
s.t. $w \;\geqslant\; 0.$

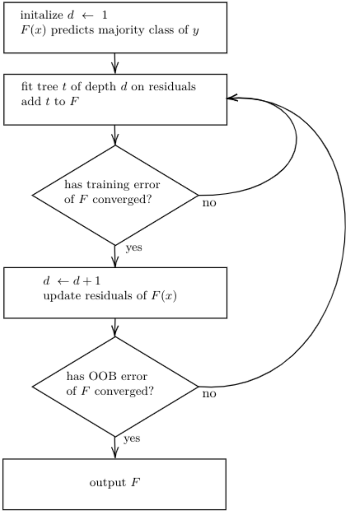


Incremental Depth Bagging

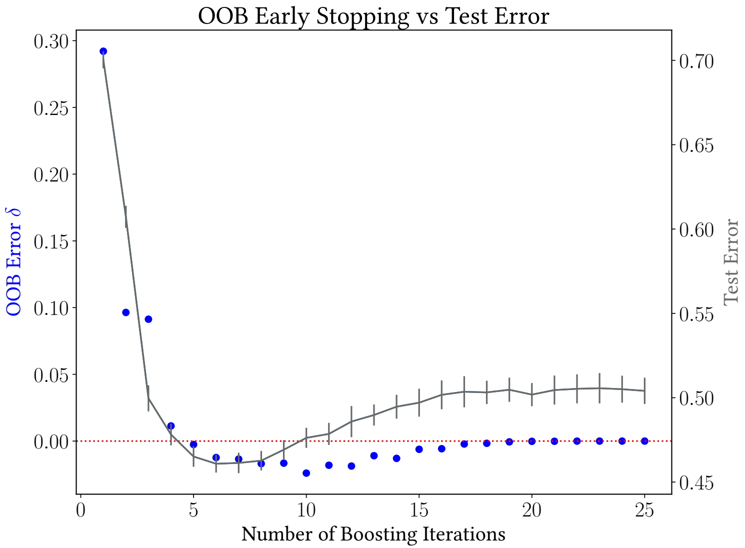


Incremental Depth Bag Boosting



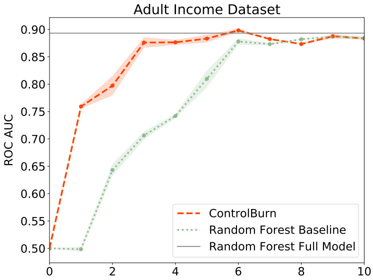


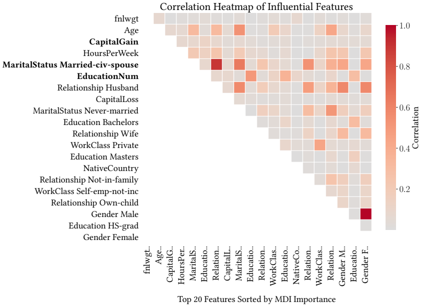
Out-of-Bag Early Stopping



Results

- ControlBurn is useful on data w/ correlated features.

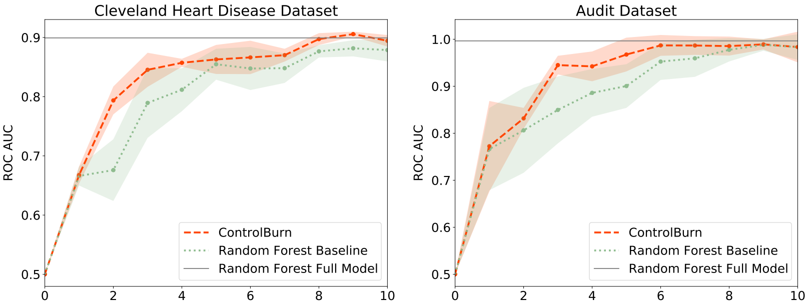




Results

- Adult income dataset: select top 3 features
- Random Forest Baseline:
  - Fnlwgt, Age, CapitalGain
  - Model AUC: **0.70**
- ControlBurn:
  - CapitalGain, MaritalStatus, EducationNum
  - Model AUC: **0.89**

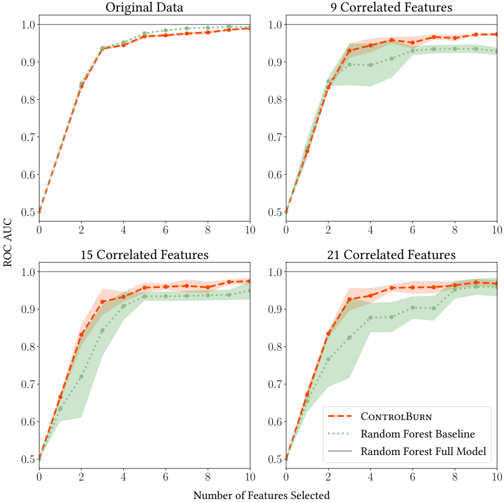
Results



Cleveland Heart Disease Dataset
Audit Dataset
ROC AUC
ControlBurn
Random Forest Baseline
Random Forest Full Model

Results

Chess dataset synthetic example:



Overfitting

- ControlBurn prevents overfitting through:
  - Explicit ℓ_1 regularization
  - Averaging predictions
  - Limiting tree depth

Conclusion

- ControlBurn uses ℓ_1 regularization to select a sparse subset of important features from a tree ensemble
- ControlBurn works best on diverse forests
- Links:
  - https://arxiv.org/abs/2107.00219
  - https://pypi.org/project/ControlBurn/
  - https://github.com/udellgroup/controlburn