ControlBurn: Feature Selection by Sparse Forests

Brian Liu, Miaolan Xie, and Madeleine Udell
Given training data $X \in \mathbb{R}^{m \times p}$ and response $y \in \mathbb{R}^{m}$

Fit a collection of base learners $T_1(x), T_2(x), \ldots, T_n(x)$ 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

$X \sim \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}$

$\bigcirc \sim \text{leaf}$

Forest $F$
Ensemble Post-Processing

- Reduces ensemble size to:
  - Prevent overfitting
  - Improve interpretability
- Friedman and Popescu (2003): $\ell_1$ post processing
- Minimize w.r.t. $\alpha$

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

$$
\text{minimize} \quad \frac{1}{m} L(y, Aw) + \lambda \sum_{i=1}^{n} u_i w_i \\
\text{subject to} \quad w \geq 0
$$

(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:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{m} L(A, w, y) + \lambda \sum_{i=1}^{n} u_i w \\
\text{s.t.} & \quad w \geq 0.
\end{align*}
\]
Incremental Depth Bagging

\[ d_{\text{max}} = 3 \]

\[ d = 1 \]

\[ d = 2 \]

\[ d = 3 \]

\[ \text{initialize } d \leftarrow 1, \ F \leftarrow \emptyset \]

\[ \text{add tree of depth } d \text{ to } F \]

\[ \text{has training error of } F \text{ converged?} \]

\[ \text{yes} \]

\[ d \leftarrow d + 1 \]

\[ \text{is } d > d_{\text{max}} \text{?} \]

\[ \text{no} \]

\[ \text{output } F \]

\[ \text{yes} \]
Incremental Depth Bag Boosting

1. Initialize $d \leftarrow 1$
2. $F(x)$ predicts majority class of $y$
3. Fit tree $t$ of depth $d$ on residuals and add $t$ to $F$
4. Compute residuals of $F_1$
5. Fit $F_2$ on residuals and compute OOB error
6. Compute residuals of $F_1 + F_2$
7. Repeat until OOB error converges
8. $d \leftarrow d + 1$
9. Update residuals of $F(x)$
10. Has OOB error of $F$ converged?
   - No
   - Yes
11. Has training error of $F$ converged?
    - No
    - Yes
12. Output $F$
Out-of-Bag Early Stopping

OOB Early Stopping vs Test Error

Number of Boosting Iterations

OOB Error $\delta$

Test Error

Number of Boosting Iterations
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
Cleveland Heart Disease Dataset

Audit Dataset
Results

Chess dataset synthetic example:

![Graphs showing ROC AUC for different numbers of features selected from original data and various correlated feature sets. The graphs compare CONTROLBURN, Random Forest Baseline, and Random Forest Full Model.]
Overfitting

- **CONTROLBURN** prevents overfitting through:
  - Explicit $\ell_1$ regularization
  - Averaging predictions
  - Limiting tree depth
Conclusion

- **ControlBurn** uses $\ell_1$ regularization to select a sparse subset of important features from a tree ensemble.
- **ControlBurn** works best on diverse forests.
- Links:
  - [https://pypi.org/project/ControlBurn/](https://pypi.org/project/ControlBurn/)
  - [https://github.com/udellgroup/controlburn](https://github.com/udellgroup/controlburn)