Random Forests

Random Forest make CART Trees more stable, accurate and performant by bagging [Bootstrap Aggregation]

RF randomizes Tree Learner and then combines many models (same fundamental structure) into one meta model
→ Homogenous ensembles

⇒ this ensembled meta model gives better predictions than a single trained Tree, but its a bit of a black box and not easy to interpret

Training bagged ensembles Train Base Learner (CART for Random Forest) on M Bootstrap samples of training data $\mathcal{D}$, with n_{train})

- each draw sample size n WITH replacement, same obs. multiple times possible
- Datapoints sampled in one iter. "in-bag" (IB) vs. not sampled "out-of-bag" (OOB)
- Nr. of Trees where i-th observation is OOB: $S_{OOB}^{(i)} = \sum_{m=1}^{M} I[i \in OOB^{[m]}]$ $P(\,) \xrightarrow{n \to \infty} 0.37$

Fit model on each bootstrapped data $\mathcal{D}^{[m]}$ to obtain iteration model $\hat{b}^{[m]}$

↓ Ensembled "model" predicts, for each datapoint, the average /aggregated predictions of the M fitted models

Regression / Decision Score (multiclass use $\hat{f}_k$): $\hat{f}^{[M]}(x) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}^{[m]}(x)$

Classification majority voting: $\hat{h}^{[M]}(x) = \arg\max_k \sum_{m=1}^{M} I[\hat{h}^{[m]}(x) = k]$

Classification probabilities through averaging: $\hat{\pi}_k^{[M]}(x) = \frac{1}{M} \sum_{i=1}^{M} \hat{\pi}_k^{[m]}(x)$

if iter. model predict labels
Prob. by class frequencies: $\hat{\pi}_k^{[M]}(x) = \frac{1}{M} \sum_{m=1}^{M} I[\hat{h}^{[m]}(x) = k]$

"Out of Bag" Error Estimation for GE

Regr: $\hat{f}_{OOB}^{(i)} = \frac{1}{S_{OOB}^{(i)}} \sum_m I(i \in OOB^{[m]}) \cdot \hat{f}^{[m]}(x^{(i)})$

⇒ $\widehat{GE}_{OOB} = \frac{1}{n} \sum_i L(y^{(i)}, \hat{f}_{OOB}^{(i)})$ not optimistically biased!

Use OOB error for first impression and Tuning of M & others
Unique for RF/bagging, wide model comparison still with CV etc.

When does Bagging help? Bagging is most useful on somewhat unstable Learners such as CART Trees to reduce variability (i.e. random errors) of predictions
where errors are mainly due to randomness rather than systematic issues

In essence it use the oldest statistical trick in the book: Run high variance (random) processes many times and average over it to **smoothen** results
smoother pred. curves & decision bound. with RF

Intuition: (Expected quadratic Loss over individ. BL predictions $b^{[m]}(x)$) ≥ (quadratic Loss of ensemble prediction $f^{[M]}(x)$)

Note that only stochastic part here is bootstrap sample in m-th tree given the training data from $\mathbb{P}_{xy}$. EV in proof refers to $\mathbb{E}_{\mathcal{D}_{train}^{[m]} | \mathcal{D}_{train} \sim \mathbb{P}_{xy}}$

$\mathbb{E}((y - b^{[m]}(x))^2) = Var(y - b^{[m]}(x)) + (\mathbb{E}(y - b^{[m]}(x)))^2 \overset{Var \geq 0}{\Longrightarrow} \mathbb{E}((y - b^{[m]}(x))^2) \geq (\mathbb{E}(y - b^{[m]}(x)))^2$ making use of Var-Verschiebungssatz (LOTV: "Law of total variance")

$\overset{y\text{ given}}{\iff} \mathbb{E}((y - b^{[m]}(x))^2) \geq (y - \underbrace{\mathbb{E}(b^{[m]}(x))}_{\text{need to show this } = f^{[M]}(x)})^2$

all M bootstr. samples drawn with equal prob. WITH replacement
$\mathbb{E}(b^{[m]}(x)) = \sum_{m=1}^{M} b^{[m]}(x) \cdot P(\mathcal{D}_{train}^{[m]}) \overset{!}{=} \frac{1}{M} \sum_{i=1}^{M} b^{[m]}(x) =: f^{[M]}(x)$

Hyperparam. 100 or 500 are reasonable default, M better but converges with diminishing returns & computing costs....

Ideally the ensemble-Variance would go down linear in $\boxed{\text{ensemble size } M}$, but that is unrealistic considering the correlation of single iter. models
Bootstrap samples are correlated by design ⟹ models $\hat{b}^{[m]}$ correlated which increases the variance of ensemble $\hat{f}$ ..... part of $GE(\hat{f})$, problematic if too large

Simplified analysis: "scalar" model(s) with const. variance, neglecting pred. error

Assuming $Var(b^{[m]}) = \sigma^2$, $Corr(\hat{b}^{[m]}, \hat{b}^{[l]}) = r$] $Var(\hat{f}^{[M]}) \approx Var\left(\frac{1}{M} \sum_{m=1}^{M} \hat{b}^{[m]}\right) = \frac{1}{M^2}\left(\sum_{m=1}^{M} Var(\hat{b}^{[m]}) + 2 \sum_{m<l} Cov(\hat{b}^{[m]}, \hat{b}^{[l]})\right) = \frac{1}{M^2}\left(M\sigma^2 + 2 \underbrace{\frac{M(M-1)}{2}}_{\text{possible comb. for } Cov()} r\sigma^2\right)$

here $= \frac{Cov(\,)}{\sigma \cdot \sigma} \iff Cov(\,) = r\sigma^2$

$= (1-r)\frac{\sigma^2}{M} + r\sigma^2$
convex mix w.r.t. r of reduction (linear) & no reduction

easy interpretation: $Var(f^{[M]}) = \frac{\sigma^2}{M}$ ↓ Linear in M, minimal for corr. = 0, $Var(f^{[M]}) = \sigma^2$ no reduction at all for r = 1 i.e. perfect correlation of bootstrap samples

Decorrelation Random Forests can decorrelate Trees with a simple randomization

mtry-baseline w.o. Tuning: $\sqrt{p}$ for classific., p/3 for Regr.
fixed number of cand. (mtry in R) is Hyperparameter

Node based (more common than for entire Tree once) ⤻ For each Node randomly draw without replacement the candidate features for splitting

We want the Trees to make **different** mistakes/errors so that the averaging yields the best performance increase. We actually then use (almost) fully expanded Trees without aggressive early stopping or pruning to increase variability of each individual tree (then $\widehat{GE}$ larger than without any decorrelation method)
But careful: Setting mtry very small results in highly decorrelated Trees, but they become too random itself! mtry ≤ p should neither be too small nor approach p.

↓ not actual MSE decomposition (nonexistent for classific. loss), more conceptual
In terms of Bias-Variance Tradeoff for Ensemble Model: Deep instable iter. Trees minimize Bias, we allow inherent high variance as we can average it out [aggregated!]
(Bias ≙ Prediction error, not random Var ≙ Prediction Instability to random Δ data) Decorrelation needs to find middle ground, if its too extrem we have problem with (random) Bias / Inaccuracy

Feature Importance Get some interpretability for RF by measuring feature contribution to ensemble model performance

Permutation: Permute feature x_j to ψ_j distorting the feature-target relation (1) Repeat this many times to avg. out random x_j-permutation (4)
Compute all n OOB pred. with ψ_j and obtain new $\widehat{GE}_{OOB,\psi_j}$ (2) ⟹ measure importance of x_j as $\widehat{GE}_{OOB,\psi_j} - \widehat{GE}_{OOB}$ (3) ⇓ $\widehat{FI}_j$ = avg. of $\widehat{GE}$-diffs
No new model fit with M× tree growing required! We simply recalculate new (OOB) predictions for same (OOB) data, just with altered x_j

"Removing" Feature: For all models $\hat{b}^{[m]}$ find all split-Nodes N that use feature x_j (1)
Look up risk improvement in N (2) ⟶ Add up all improvement across all N and all $\hat{b}^{[m]}$ (3)

Proximities Random Forests have built-in similarity measure for pairs of observations i, j

After training push all observation through each forrest tree

Symmetric $n \times n$ proximity matrix with $\text{prox}(x^{(i)}, x^{(j)}) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}[\text{leaf}(i;m) = \text{leaf}(j;m)]$

$\hat{=}$ percentage of how often both datapoints are placed in same terminal node of a tree

$$\begin{array}{cc|cccc} & & & \text{obs.} & & \\ & & 1 & 2 & \cdots & n \\ \hline & 1 & 1 & 0.3 & \cdots & 0.5 \\ & 2 & 0.3 & 1 & & \\ \text{obs.} & \vdots & \vdots & & \ddots & \\ & n & 0.5 & & & 1 \end{array}$$

We can visualize matrix by projecting it into lower dim space, e.g. via multidim. scaling (might have to turn proximities into distances Distance = 1 − proximity)

$\hookrightarrow$ inspection of within-class variance and class overlaps

Note: Observation from same class should usually form identifiable clusters

Proximity matrix used for u.a.

- clustering

- **outlier detection**, i.e. observation with small prox( ) to all others
- Identifying mislabel data (manually labeled or ambigiously labeled) as proximity outliers

× **Imputing missing data** on new unseen data for RF at pred. time

(1) Replace NAs in each feature with the median of available values

(2) Compute proximities (was impossible with NAs)

(3) Replace missings per feature by $\underbrace{\text{weighted average / majority vote}}_{\text{of all non missings}}$, weights proportional to proximity