Random Forests

by Jaehyeong Ahn

Department of Applied Statistics, Konkuk University

jayahn0104@gmail.com

Random Forests

Topics to be covered

- Random forest recipe
- Why do random forests work?
- Ramifications of random forests
    - Out-of-bag error estimate
    - Variable importance
    - Proximity
    - Missing value imputation
    - Outliers
    - Unsupervised learning
    - Balancing prediction error

Random Forest Recipe

Notations

- $\mathfrak{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ be the data
    - $x^{(i)} \in \mathfrak{X}$ and $y^{(i)} \in \mathfrak{Y}$
    - $\mathfrak{X}$ is the input space with p features and $\mathfrak{Y}$ is the output space
    - $\mathfrak{Y} = \mathbb{R}^n$, in regression
    - $\mathfrak{Y}$ is a discrete finite set of K elements, in classification
    - $\mathfrak{D}(k)$: a bootstrapped sample, called the k-th "in-bag" set
    - OOB(k): the set of data points not in $\mathfrak{D}(k)$, called the k-th "out-of-bag" set

$$\mathfrak{D} = \mathfrak{D}(k) \cup OOB(k), \quad \mathfrak{D}(k) \cap OOB(k) = \emptyset$$

- Regulate a parameter
    - $m \ll p$
- Fix some constants
    - N_{min}
    - B

Random Forest Recipe

The way Random Forest works

Do for $k = 1$ to B

- Draw a bootstrap resample $\mathfrak{D}(k)$ from $\mathfrak{D}$
- Grow a tree T_k using $\mathfrak{D}(k)$ by applying the following splitting rule:
    - At each node, randomly select m features (out of total p features)
    - Split the node by choosing the split that minimizes the empirical risk among all possible splits using only these m selected features
    - Stop splitting the node if the terminal node contains fewer than N_{min} elements. Where N_{min} is a predetermined number
    - Do not prune
- From each tree T_k, get the predictor φ_k for $k = 1, \cdots, B$

Get the final predictor $\zeta(x)$:

- For regression:

$$\zeta(x) = Av_k\{\varphi_k(x)\} = \frac{1}{B}\sum_{k=1}^{B}\varphi_k(x)$$

- For classification:

$$\zeta(x) = Plur_k\{\varphi_k(x)\} = \underset{j\in\mathfrak{Y}}{\operatorname{argmax}}\,|\{k : \varphi_k(x) = j\}|$$

Why do random forests work?

Case A: Regression

- Preparation, Before get into the main idea
    - Let us write $\varphi_k(x)$ gotten out of $\mathfrak{D}(k)$ as $\varphi(x, \theta_k)$
    - θ_k stands for all those things that were involved in the construction of $\varphi_k(x)$
    - Treating θ as a random variable, we write $\varphi(x, \theta)$
    - We don't need to specify the distribution of θ as it is used only a theoretical background
    - But we can talk about the expectation $E_\theta \varphi(x, \theta)$ of $\varphi(x, \theta)$ with respect to θ and the probability $P_\theta(\varphi(X, \theta) = j)$

- Define an estimator of $E_\theta \varphi(x,\theta)$

$$AV_k\varphi_k(x) \to E_\theta\varphi(x,\theta), \quad (AV_k\varphi_k(x) = \tfrac{1}{B}\textstyle\sum_{k=1}^{B}\varphi(x,\theta_k))$$

- Define the Prediction Error(risk) of the forest

$$PE^*(forest) = E|Y - E_\theta\varphi(Y,\theta)|^2, \quad (E \text{ stands for } E_{X,Y})$$

- Define the average(expected) Prediction Error of individual tree

$$PE^*(tree) = E_\theta E|Y - \varphi(X,\theta)|^2$$

- Assume the Unbiasedness of forest and tree(strong assumption)

$$E[Y - E_\theta\varphi(x,\theta)] = 0, \quad E[Y - \varphi(X,\theta)] = 0, \forall\theta$$

- Express $PE^*(forest)$ using Covariance

$$\begin{aligned} PE^*(forest) &= E|Y - E_\theta \varphi(X,\theta)|^2 \\ &= E|E_\theta[Y - \varphi(X,\theta)]|^2, \quad (\because Y \text{ is independent from } \theta) \\ &= E\{E_\theta[Y - \varphi(X,\theta)] \cdot E_{\theta'}[Y - \varphi(X,\theta^{'})]\} \\ &= EE_{\theta,\theta'}(Y - \varphi(X,\theta)) \cdot (Y - \varphi(X,\theta^{'})) \\ &= E_{\theta,\theta'}E(Y - \varphi(X,\theta)) \cdot (Y - \varphi(X,\theta^{'})) \\ &= E_{\theta,\theta'}Cov\left(Y - \varphi(X,\theta), Y - \varphi(X,\theta^{'})\right) \end{aligned}$$

Where $\theta, \theta^{'}$ are independent with the same distribution

- Using the covariance-correlation formula

$$\begin{aligned} & Cov\left(Y - \varphi(x,\theta), Y - \varphi(x,\theta^{'})\right) \\ &= \rho\left(Y - \varphi(x,\theta), Y - \varphi(x,\theta^{'})\right) \cdot std\left(Y - \varphi(x,\theta)\right) \cdot std\left(Y - \varphi(x,\theta^{'})\right) \\ &= \rho(\theta,\theta^{'})sd(\theta)sd(\theta^{'}) \end{aligned}$$

Where $\rho(\theta,\theta^{'})$ is the correlation between $Y - \varphi(x,\theta)$ and $Y - \varphi(x,\theta^{'})$,

$sd(\theta)$ denotes the standard deviation of $Y - \varphi(x,\theta)$

- Express $PE^*(forest)$ using correlation

$$PE^*(forest) = E_{\theta,\theta'}\{\rho(\theta,\theta^{'})sd(\theta)sd(\theta^{'})\}$$

- Define the weighted average correlation $\bar{\rho}$

$$\bar{\rho} = \frac{E_{\theta,\theta'}\{\rho(\theta,\theta^{'})sd(\theta)sd(\theta^{'})\}}{E_\theta sd(\theta)E_{\theta'} sd(\theta')}$$

- Show the inequality between $PE^*(forest)$ and $PE^*(tree)$ by the property of Variance

$$\begin{aligned} PE^*(forest) &= \bar{\rho}(E_\theta sd(\theta))^2 \\ &\leq \bar{\rho}E_\theta(sd(\theta))^2 \\ &= \bar{\rho}E_\theta E|Y - \varphi(X,\theta)|^2, \quad (\because E[Y - \varphi(X,\theta)] = 0,\ \forall\theta) \\ &= \bar{\rho}PE^*(tree) \end{aligned}$$

- What this inequality means

"the smaller the correlation, the smaller the prediction error."

⇒ This shows why Random forest limits the number of variables $m \ll d$ at each node because it's reasonable to assume that two sets of features that have not much overlap between them should in general have a small correlation

Case A: Regression (Simplified version)

- Prior Assumption
  - It is empirically verified that the random forests have very small bias
  - So the empirical risk is basically captured by the variance
- Assumption
  - Assume that $\varphi_1, \cdots, \varphi_B$ as random variables which have the same law with finite standard deviation σ
  - Let ρ be the correlation between any of them$(\varphi_1, \cdots, \varphi_B)$

Variance of predictors

$$Var\left(\frac{1}{B}\sum_{k=1}^{B}\varphi_k\right) = \frac{1}{B^2}\sum_{k,m}Cov(\varphi_k,\varphi_m)$$

$$= \frac{1}{B^2}\sum_{k\neq m}Cov(\varphi_k,\varphi_m) + \frac{1}{B^2}\sum_{k}Var(\varphi_k)$$

$$= \frac{1}{B^2}(B^2 - B)\rho\sigma^2 + \frac{1}{B^2}B\sigma^2$$

$$= \rho\sigma^2 + \frac{\sigma^2}{B}(1-\rho)$$

What it means

- The first term $\rho\sigma^2$ is small if the correlation ρ is small, which can be achieved by $m \ll$ *d*
- The second term is small if B is large. i.e. if we take a large number of trees

Case B: Classification

- Define the estimator of $\underset{j}{\text{argmax}}\, P_\theta(\varphi(X,\theta) = j)$

$$\underset{j}{\text{argmax}}\, \frac{1}{B}|\{k : \varphi(X,\theta_k) = j\}| \to \underset{j}{\text{argmax}}\, P_\theta(\varphi(X,[theta) = j)$$

- Define the Margin Function

$$mr(X,Y) = P_\theta(\varphi(X,\theta) = Y) - \max_{j \neq Y} P_\theta(\varphi(X,\theta) = j)$$

  - in the limit as $k \to \infty$
    - (i) if $mr(X,Y) \geq 0$, then the R.F classifier ζ will predict that the class label of X is Y
    - (ii) if $mr(X,Y) < 0$, then the R.F classifier ζ will predict that the class label of X is something other than Y
  - Hence, the classification error occurs when and only when $mr(X,Y) < 0$ (ignore the tie case)

- Define the Prediction Error of forest classifier using $mr(X, Y)$

$$PE^* = P(mr(X, Y) < 0)$$

- Define the mean(expectation) s of $mr(X, Y)$

$$s = E[mr(X, Y)]$$

s represents the strength of the individual classifiers in the forest

- **(Lemma 1)** Let U be a random variable and let s be any positive number. Then

$$Prob[U < 0] \leq \frac{E|U - s|^2}{s^2}$$

proof: by the Chebyshev inequality

- Express the inequality of Prediction Error by **Lemma 1**

$$PE^* \leq \frac{Var(mr)}{s^2}, \quad \text{(assume } s > 0\text{)}$$

- Define $\hat{j}(X,Y)$ for simple notation

$$\hat{j}(X,Y) = \operatorname*{argmax}_{j \neq Y} P_\theta(\varphi(x,\theta) = j)$$

- Express $mr(X,Y)$ using $\hat{j}(X,Y)$

$$\begin{aligned} mr(X,Y) &= P_\theta[\varphi(X,\theta) = Y] - P_\theta[\varphi(X,\theta) = \hat{j}(X,Y)] \\ &= E_\theta[I(\varphi(X,\theta) = Y) - I(\varphi(X,\theta) = \hat{j}(X,Y))] \end{aligned}$$

- Define $rmg(\theta, X, Y)$ for simple notation

$$rmg(\theta, X, Y) = I(\varphi(X,\theta) = Y) - I(\varphi(X,\theta) = \hat{j}(X,Y))$$

- Hence we can express $mr(X,Y)$ using $rmg(\theta, X, Y)$

$$mr(X,Y) = E_\theta[rmg(\theta, X, Y)]$$

- Express $Var(mr)$ by Covariance

$$
\begin{aligned}
&Var(mr)\\
&= E(E_\theta rmg(\theta, X, Y))^2 - (EE_\theta rmg(\theta, X, Y))^2\\
&= EE_\theta rmg(\theta, X, Y)E_{\theta'} rmg(\theta^{'}, X, Y) - EE_\theta rmg(\theta, X, Y)EE_{\theta'} rmg(\theta^{'}, X, Y)\\
&= EE_{\theta,\theta'}\{rmg(\theta, X, Y)rmg(\theta^{'}, X, Y)\} - E_\theta Ermg(\theta, X, Y)E_{\theta'} Ermg(\theta^{'}, X, Y)\\
&= E_{\theta,\theta'}\{E[rmg(\theta, X, Y)rmg(\theta^{'}, X, Y)] - Ermg(\theta, X, Y)Ermg(\theta^{'}, X, Y)\}\\
&= E_{\theta,\theta'} Cov\left(rmg(\theta, X, Y), rmg(\theta^{'}, X, Y)\right)
\end{aligned}
$$

- Using the Covariance - Correlation formula

$$Cov\left(rmg(\theta, X, Y), rmg(\theta^{'}, X, Y)\right)$$

$$
\begin{aligned}
&= \rho\left(rmg(\theta, X, Y), rmg(\theta^{'}, X, Y)\right) \cdot std(rmg(\theta, X, Y) \cdot std(rmg(\theta^{'}, X, Y))\\
&= \rho(\theta, \theta^{'})sd(\theta)sd(\theta^{'})
\end{aligned}
$$

Where $\rho(\theta, \theta^{'})$ is the correlation between $rmg(\theta, X, Y)$ and $rmg(\theta^{'}, X, Y)$,
$sd(\theta)$ is the standard deviation of $rmg(\theta, X, Y)$

- Express $Var(mr)$ using correlation

$$Var(mr) = E_{\theta,\theta'}[\rho(\theta, \theta^{'})sd(\theta)sd(\theta^{'})]$$

- Define the weighted average correlation $\bar{\rho}$ of $rmg(\theta, X, Y)$ and $rmg(\theta^{'}, X, Y)$

$$\bar{\rho} = \frac{E_{\theta,\theta'}[\rho(\theta, \theta^{'})sd(\theta)sd(\theta^{'})]}{E_{\theta,\theta'}[sd(\theta)sd(\theta')]}$$

- Show the inequality of $Var(mr)$ using the fact that $|\bar{\rho}| \leq 1$ and $sd(\theta) = sd(\theta^{'})$

$$Var(mr) = \bar{\rho}\{E_{\theta} sd(\theta)\}^2 \leq \bar{\rho} E_{\theta}[sd(\theta)]^2$$

- Unfold $E_{\theta}[sd(\theta)]^2$

$$E_{\theta}[sd(\theta)]^2 = E_{\theta} E[rmg(\theta, X, Y)]^2 - E_{\theta}[Ermg(\theta, X, Y)]^2$$

- Show the inequality of s by ...

$$s = E(mr(X, Y)) = EE_{\theta} rmg(\theta, X, Y) = E_{\theta} Ermg(\theta, X, Y)$$

$$\begin{aligned} s^2 &= \{E_{\theta} Ermg(\theta, X, Y)\}^2 \\ &\leq E_{\theta}[Ermg(\theta, X, Y)]^2 \end{aligned}$$

- Therefore

$$E_{\theta}[sd(\theta)]^2 \leq E_{\theta} E[rmg(\theta, X, Y)]^2 - s^2$$

- Show the inequality of $E_{\theta} E[rmg(\theta, X, Y)]^2$

$$E_{\theta} E[rmg(\theta, X, Y)]^2 \leq 1$$

$$(\because rmg(\theta, X, Y) = I(something) - I(somethingelse),$$

Which can only take 0 or ±1 for its value)

- Show the inequality of Prediction Error

$$PE^* \leq \bar{\rho}\frac{1-s^2}{s^2}$$

- Above inequality can be proven by the inequalities we've shown before

1. $$PE^* \leq \frac{Var(mr)}{s^2}$$

2. $$Var(mr) = \bar{\rho}\{E_\theta sd(\theta)\}^2 \leq \bar{\rho}E_\theta[sd(\theta)]^2$$

3. $$E_\theta[sd(\theta)]^2 \leq E_\theta E[rmg(\theta, X, Y)]^2 - s^2$$

4. $$E_\theta E[rmg(\theta, X, Y)]^2 \leq 1$$

- What this inequality means

"the smaller the correlatoin, the smaller the prediction error"

Ramifications of RF

We now show how to exploit the RF to extract some important side information

Out-of-Bag(OOB) error estimate

- Recall that at stage k, for $k = 1, \cdots, B$, of the RF construction, roughly 1/3 of $\mathfrak{D}$ is left out from the bootstrap sample $\mathfrak{D}(k)$, and this left-out set is denoted by $OOB(k)$
  - As we wrote before, $\mathfrak{D}$ is a disjoint union of $\mathfrak{D}(k)$ and $OOB(k)$
- For each k, using $\mathfrak{D}(k)$, construct a tree T_k which a predictor $\varphi_k(x)$ is derived
- **(Idea)** Since $OOB(k)$ is not used in the construction of $\varphi_k(x)$, it can be used to test $\varphi_k(x)$

  $\Rightarrow$ So OOB error can be obtained as a proxy for the generalization error

Procedure

- For each data point $x^{(i)} \in OOB(k)$, compute $\varphi_k(x^{(i)})$ only if $x^{(i)}$ is in $OOB(k)$ for all $k = 1, \cdots, B$
- Aggregate the OOB predictors $\varphi_k(x^{(i)})$ for only **some** $k = 1, \cdots, B$. Since we only consider the OOB sets which have $x^{(i)}$
    - For regression:
    $$\alpha(x^{(i)}) = Av_k\{\varphi_k(x^{(i)})\}$$
    - For classification:
    $$\alpha(x^{(i)}) = Plur_k\{\varphi_k(x^{(i)})\}$$
- To compute $\alpha(x^{(i)})$ for all $i = 1, \cdots, N \quad B$ must be big enough
    - Since the probability of any single data point belonging in all of $\mathfrak{D}(k)$ for $k = 1, \cdots, B$ is very small if B is big enough
- Obtain the overall error(OOB error estimate) of the random forest
$$OOB_{error} = \sum_{\{x^{(i)} \in OOB(k)\}_1^B} L(y_i, \alpha(x^{(i)})$$
    - In regression: $L(y_i, \alpha(x^{(i)})$ denotes a square loss function
    - In classification: $L(y_i, \alpha(x^{(i)}))$ denotes a misclassification error

⇒ We may regard OOB_{error} as a reasonably good proxy for the generalization error

Variable importance

Impurity importance

- Recall that in the RF recipe at each node the split decision is made to make the impurity(empirical risk) decrease the most
- **(Idea)** At that node, it is reasonable to presume the variable involved with the split decision is the most important one
- The **impurity importance** of each variable is the addition of the absolute values of such decrease amounts for all splits in all trees in the forest every time the split is done using *that* variable
- Notations
  - $\mathbb{T}_j$ is the set of all nodes split by j-th variable
  - $\varDelta R_{nt}$ denotes the decrease amounts of empirical risk at node t
- Define the impurity variable importance of j-th feature

$$VI(j) = \sum_{t \in \mathbb{T}_j} |\varDelta R_{nt}|$$

Permutation importance

- The permutation importance assumes that the data value associated with an important variable must have a certain structural relevance to the problem
- **(Idea)** So that the effect of a random shuffling (permutation) of its data value must be reflected in the decrease in accuracy of the resulting predictor
- Notations
  - $\mathfrak{D} = \{x^{(1)}, \cdots, x^{(n)}\}$
  - $x^{(i)} = [x_1^{(i)}, \cdots, x_p^{(i)}]^T$
- Write $\mathfrak{D}$ in matrix form

$$\mathfrak{D} = \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{pmatrix}$$

in which $x_{ij} = x_j^{(i)}$ for all $i = 1, \cdots, n$ and $j = 1, \cdots, p$

- Let π be a permutation on $\{1, \cdots, n\}$ denoted by

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{pmatrix}$$

- Define the permuted data set $\mathfrak{D}(\pi, j)$ gotten from $\mathfrak{D}$ by applying π to the j-th column

$$\mathfrak{D}(\pi, j) = \begin{pmatrix} x_{11} & \cdots & x_{\pi(1)j} & \cdots & x_{1p} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \cdots & x_{\pi(i)j} & \cdots & x_{ip} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots & x_{\pi(n)j} & \cdots & x_{np} \end{pmatrix}$$

$\Rightarrow$In $\mathfrak{D}(\pi, j)$ the values of the j-th column in $\mathfrak{D}$ is shuffled by the permutation π

- So, if x_j is important, the error using the randomly shuffled dataset $\mathfrak{D}(\pi, j)$ should be much worse that that using the correct dataset $\mathfrak{D}$
- On the other hand, a variable has not much relevance or importance, if the error using $\mathfrak{D}(\pi, j)$ is more or less the same as the one with $\mathfrak{D}$

Permutation importance by OOB set

Classification

- For each k, let φ_k be the predictor associated with the tree T_k
  Let π_k be a permutation on the k-th out-of-bag set $OOB(k)$
- Define $C(\pi, j, k)$

$$C(\pi, j, k) = \sum_{(x^{(i)}, y^{(i)}) \in OOB(k)} \left\{ I\left(y^{(i)} = \varphi_k(x^{(i)})\right) - I\left(y^{(i)} = \varphi_k(x^{(i)}(\pi_k, j))\right) \right\}$$

- Define the permutation variable importance of the j-th variable x_j in k-th tree

$$VI(j, k) = \frac{C(\pi_k, j, k)}{|OOB(k)|}$$

- Define the permutation variable importance of the j-th variable x_j in forest

$$VI(j) = \frac{1}{B} \sum_{k=1}^{B} VI(j, k)$$

⇒ It is very reasonable to presume that the higher $VI(j)$ is, the more important x_j is

Regression

- For each k, let φ_k be the regressor associated with the tree T_k Let π_k be a permutation on the k-th out-of-bag set $OOB(k)$
- Define $S(k)$ which denotes the empirical risk of OOB set with square loss

$$S(k) = \sum_{(x^{(i)},y^{(i)}) \in OOB(k)} |y^{(i)} - \varphi_k(x^{(i)})|^2$$

- Define $S(\pi, j, k)$ which is the empirical risk of the permuted OOB set in which $x^{(i)}$ is replaced with $x^{(i)}(\pi_k, j)$

$$S(\pi, j, k) = \sum_{(x^{(i)},y^{(i)}) \in OOB(k)} |y^{(i)} - \varphi_k(x^{(i)}(\pi_k, j))|^2$$

- Define the permutation variable importance of the j-th variable x_j in k-th tree

$$VI(j, k) = \frac{S(\pi_k, j, k)}{S(k)}$$

- Define the permutation variable importance of the j-th variable x_j in forest

$$VI(j) = \frac{1}{B} \sum_{k=1}^{B} VI(j, k)$$

⇒the higher $VI(j)$ is, the more important x_j is

Proximity

- The proximity of two data points is as measure of how close they are
- Knowledge of the proximity for all data points, if available, is very important information that can be exploited in many ways (e.g. nonlinear dimension reduction, clustering problems, $\cdots$)
- However defining the proximity of categorical varaible is some what challenging. And also in numeric data, normalizing the unit of measurement is not that simple issue (e.g. meters, millimeters, kilometers, $\cdots$)
- RF provides a handy and sensible way of defining the proximity
- RF defines the proximity of two data points using terminal nodes
- **(Idea)** If two data points end up in the same terminal node, then they are close (the proximity is 1)

- Recall that RF constructs a tree T_k for each bootstrap sample $\mathfrak{D}(k)$ for $k = 1, \cdots, B$. And from T_k, k-th predictor φ_k is derived
- Let $v_k(x)$ be the terminal node of x in k-th tree T_k
    - Given any input x, one can put it down the tree T_k to eventually arrive at a terminal node which is denoted by $v_k(x)$
- Two input points with the same terminal nodes have the same $\varphi_k(x)$ values but the converse does not hold
- Define the k-th proximity $Prox_k(i, j)$ of two data points $x^{(i)}$ and $x^{(j)}$, regardless of whether in-bag or out-of-bag for each $k = 1, \cdots, B$, and for $i, j = 1, \cdots, N$

$$Prox_k(i, j) = \begin{cases} 1 & \text{if } v_k(x^{(i)}) = v_k(x^{(j)}) \\ 0 & \text{if } v_k(x^{(i)}) \neq v_k(x^{(j)}) \end{cases}$$

$\Rightarrow$ this says that $Prox_k(i, j) = 1$ if and only if $x^{(i)}$ and $x^{(j)}$ end up in the same terminal node after the tree run on T_k

- Define the **Proximity** of forest

$$Prox(i, j) = \frac{1}{B} \sum_{k=1}^{B} Prox_k(i, j)$$

    - Note that $Prox(i, i) = 1$ and $0 < Prox(i, j) \leq 1$

Missing value imputation

- **Mean/mode replacement (rough fill)**

  This is a fast and easy way to replace missing values

  - If x_j is numeric, then take as the imputed value the mean or median of the non-missing vlaues of the j-th feature
  - If x_j is categorical, then take the most frequent value (majority vote) of the j-th feature

- **Proximity-based imputation**
  1. Do a rough fill (mean/mode imputation) for missing input values
  2. Construct a RF with the imputed input data and compute the proximity using this RF
  3. Let $x_{ij} = x_j^{(i)}$ be a missing value in the original dataset (before any imputation)
     - Case A: x_{ij} is numerical

       Take the weighted average of the non-missing vlaues weighted by the proximity

       $$x_{ij} = \frac{\sum_{k \in \mathbb{N}_j} Prox(i,k) x_{kj}}{\sum_{k \in \mathbb{N}_j} Prox(i,k)},$$

       where $\mathbb{N}_j = \{k : x_{kj} \text{ is non-missing}\}$
     - Case B: x_{ij} is categorical

       Replace x_{ij} with the most frequent value where the frequency is calculated with the proximity as weights

       $$x_{ij} = \underset{\ell}{\operatorname{argmax}} \{ \sum_{k \in \mathbb{N}_j} Prox(i,k) I(x_{kj} = \ell) \}$$
  4. Repeat (2), (3) several times (typically 4 $\sim$ 6 times)

Outliers

- **(Idea)** An outlier in a classification problem is a data point that is far away from the rest of the data with the same output label
- To measure it, define $\bar{P}(i)$ for the i-th data point

$$\bar{P}(i) = \sum_{k \in \mathbb{L}(i)} Prox^2(i,k),$$

Where $\mathbb{L}(i) = \{k : y^{(k)} = y^{(i)}\}$ is the set of indices whose output label is the same as that of $y^{(i)}$

- Define the raw outlier measure $mo(i)$ for the i-th data point

$$mo(i) = N/\bar{P}(i)$$

This measure is big if the average proximity is small, i.e. the i-th data point is far away from other data points with the same label

- Define the final outlier measure
    - Let $\mathbb{N}_\ell = \{i : y^{(i)} = \ell\}$, where ℓ denotes an output label

$$fmo(i) = \frac{mo(i) - median\left(mo(i)I(i \in \mathbb{N}_\ell)\right)_1^N}{\sum_{i \in \mathbb{N}_\ell}\left((mo(i) - median(mo(i)I(i \in \mathbb{N}_\ell))_1^N\right)}$$

Unsupuervised learning

- Unsupervised learning by definition deals with data that has no y-part, i.e. no output labels.
    - So the dataset is of the form $\mathfrak{D} = \{x^{(i)}\}_{i=1}^{N}$
- One of the goals of unsupervised learning is to discover how features are correlated to each other
- **(Idea)** So it is a good idea to compare the given dataset with one whose features are uncorrelated to each other

Procedure

- Assign label 1 to every data point in the original dataset $\mathfrak{D}$

$$\mathfrak{D}_1 = \{(x^{(i)}, 1)\}_{i=1}^{N}$$

- Create anotehr dataset $\mathfrak{D}_2$ as follows
    - For each feature j, randomly select $x_j^{'}$ from the set of possible values the j-th feature can take
    - Repeating for $j = 1, \cdots, p$, form a vector $x^{'} = (x_1^{'}, \cdots, x_p^{'})$

      $\Rightarrow$ In this vector $x^{'}$, any dependency or correlation among features is destroyed
    - Assign label 2 to this vector. Do this N times to form a new dataset

$$\mathfrak{D}_2 = \{(x^{'(i)}, 2)\}_{i=1}^{N}$$

- Merge these two datasets to form a new bigger dataset

$$\mathfrak{D} = \mathfrak{D}_1 \cup \mathfrak{D}_2$$

What we can do with new dataset $\mathfrak{D}$

- With new dataset $\mathfrak{D}$, we can do supervised learning using a RF and get some useful information
    - Caculate the OOB error estimate
        - If the OOB error rate is big, say 40%, this method basically fails and there is not much to say
        - If, on the other hand, this error rate is low, it is reasonable to presume that the dependency structure of class 1 has **some meaning**
    - And one can compute the proximity and with it one can do many things like clustering, dimension reduction and so on

Balancing prediction error

- In some data sets, some labels occur quite frequently while other labels occur rather infrequently (e.g. fraud detection, rare disease diagnosing, $\cdots$)
- In this kind of highly unbalanced data, the prediction error imbalance between classes may become a very important issue
- Since the most commonly used classification algorithms aim to minimize the overall error rate, it tends to focus more on the prediction accuracy of the majority class which results in poor accuracy for the minority class
- Two suggested methods to tackle the problem of imbalanced data using RF:
  - Balanced random forest (BRF) based on a sampling technique
  - Weighted random forest (WRF) based on cost sensitive learning

BRF

- In learning extremely imbalanced data, there is a significant probability that a bootstrap sample contains few or even none of the minority class
- To fix this problem we use a stratified bootstrap; i.e., sample with replacement from within each class
- Notation
  - m denotes a minority class
  - M denotes a majority class
  - $D_m = \{(x^{(i)}, y^{(i)}) : y^{(i)} = m\}_{i=1}^{N}$
  - $D_M = \{(x^{(i)}, y^{(i)}) : y^{(i)} = M\}_{i=1}^{N}$
- Stratified bootstrap:
  1. Draw a bootstrap sample from the minority class

     $$D_1(k) \text{ from } D_m$$

  2. Randomly draw the same number of cases from the majority class

     $$D_2(k) \text{ from } D_M \text{ for } \#D_1(k)$$

  3. Union two bootstrap samples

     $$D(k) = D_1(k) \cup D_2(k)$$

WRF

- WRF places a heavier penalty on misclassifying the minority class since the RF classifier tends to be biased towards the majority class

- The class weight are incorporated into the RF algorithm in two places
  - In the tree induction procedure

    Class weights are used to weight the impurity for finding splits

  - In the terminal nodes of each tree

    Class weights are used to calculate the "weighted majority vote" for prediction in each terminal node

- Class weights are an essential tuning parameter to achieve desired performance

WRF example

- Experimental setting
  - Let there are two classes: M and m
  - Suppose there are 70 M cases and 4 m cases
  - Assign class weights 1 to M and w to m (for example, $w = 10$)
- Unweighted probability at this node

$$\text{Probability of } M = \frac{70}{70+4}$$

$$\text{Probability of } m = \frac{4}{70+4}$$

- Weighted probability

$$\text{Probability of } M = \frac{70}{70+4*w}$$

$$\text{Probability of } m = \frac{4*w}{70+4*w}$$

- Gini impurity of this node using the unweighted probability

$$p(1-p) = \frac{70}{74} * \frac{4}{74} \approx 0.05$$

- Gini impurity using the weighted probability

$$p(1-p) = \frac{70}{110} * \frac{40}{110} \approx 0.23$$

References

1 Breiman, Leo. "Random forests." Machine learning 45, no. 1 (2001): 5-32.

2 Breiman, L. and Cutler, A., Random Forests, https://www.stat.berkeley.edu/∼breiman/RandomForests/cc_home.htm

3 Hyeongin Choi, Lecture 10: Random Forests, http://www.math.snu.ac.kr/ hichoi/machinelearning/lecturenotes/RandomForests.pdf

4 Chen, Chao Breiman, Leo. (2004). Using Random Forest to Learn Imbalanced Data. University of California, Berkeley. https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf