randomForestSRC: Multivariate Splitting Rule Vignette

Hemant Ishwaran, Fei Tang, Min Lu and Udaya B. Kogalur

Introduction

The package is capable of constructing multivariate trees for multivariate outcomes. These can either be continuous, categorical, or a mixture of the two — the nature of the outcomes informs the code as to the type of multivariate forest to be grown. When requested, performance measures can be returned, including variable importance for each Y-outcome and for each X-feature and out-of-sample error performance for each Y-outcome. Throughout we use $\mathbf{Y} = (Y_1, \ldots, Y_p)^T$ to denote the p-dimensional multivariate outcome. For simplicity we take the first $1 \leq j \leq q$ coordinates to be continuous and the remaining coordinates $q+1 \leq j \leq p$ to be categorical (with suitable modifications if all coordinates are continuous or all coordinates are categorical).

By default, multivariate trees are grown using a normalized composite splitting rule [1]. The advantage of this rule is computational speed. A disadvantage is that it's an additive independence rule that works over the Y coordinates separately, and as such it does not take into account correlation. To address this, the package now also includes a new multivariate splitting rule using Mahalanobis distance. We begin by first discussing the default composite rule. After this, we discuss the new Mahalanobis distance rule.

Regression

To describe the multivariate composite splitting rule, we begin by considering univariate regression. Thus $q = 1$ and Y is a scalar continuous outcome value. For notational simplicity, let us suppose the node t we are splitting is the root node based on the full sample size n. Let X be the feature used to split t, where for simplicity we assume X is ordered or numeric. Let s be a proposed split for X that splits t into left and right daughter nodes $t_L := t_L(s)$ and $t_R := t_R(s)$, where $t_L = \{X_i \leq s\}$ and $t_R = \{X_i > s\}$. Letting $Y_1, \ldots, Y_n$ denote the scalar outcomes in t, the mean-squared error (mse) split-statistic is

$$D(s,t) = \frac{1}{n} \sum_{i \in t_L} (Y_i - \overline{Y}_{t_L})^2 + \frac{1}{n} \sum_{i \in t_R} (Y_i - \overline{Y}_{t_R})^2$$

where $\overline{Y}_{t_L}$ and $\overline{Y}_{t_R}$ are the sample means for t_L and t_R respectively. The best split for X is the split-point s minimizing $D(s,t)$.

To extend mse splitting to the multivariate case $q > 1$, we apply the mse split-statistic to each coordinate separately. Let $\mathbf{Y}_i = (Y_{i,1}, \ldots, Y_{i,q})^T$ denote the q-dimensional multivariate outcomes, $i = 1, \ldots, n$. Then the multivariate regression split rule is

$$D_q(s,t) = \sum_{j=1}^{q} \left\{ \sum_{i \in t_L} (Y_{i,j} - \overline{Y}_{t_{L_j}})^2 + \sum_{i \in t_R} (Y_{i,j} - \overline{Y}_{t_{R_j}})^2 \right\}$$

where $\overline{Y}_{t_{L_j}}$ and $\overline{Y}_{t_{R_j}}$ are the sample means of the j-th response coordinate in the left and right daughter nodes. The goal is to minimize $D_q(s,t)$.

It is important to recognize that the multivariate splitting rule can only be effective if each outcome coordinate is measured on the same scale, otherwise we could have a coordinate j, with say very large values, and its contribution

would dominate $D_q(s,t)$. We therefore calibrate $D_q(s,t)$ by assuming that each coordinate has been standardized according to

(1)

$$\frac{1}{n}\sum_{i\in t} Y_{i,j} = 0, \quad \frac{1}{n}\sum_{i\in t} Y_{i,j}^2 = 1, \quad 1 \le j \le q.$$

The standardization is applied prior to splitting a node. To make this standardization clear, we denote the standardized responses by $Y_{i,j}^*$. With some elementary manipulations, it can be verified that minimizing $D_q(s,t)$ is equivalent to maximizing

(2)

$$D_q^*(s,t) = \sum_{j=1}^{q} \left\{ \frac{1}{n_L}\left(\sum_{i\in t_L} Y_{i,j}^*\right)^2 + \frac{1}{n_R}\left(\sum_{i\in t_R} Y_{i,j}^*\right)^2 \right\}$$

where n_L and n_R denote the sample sizes for the daughter nodes, t_L and t_R.

Classification

For multivariate classification, an averaged standardized Gini splitting rule is used. First consider the univariate case (i.e., the multiclass problem) where the outcome Y_i is a class label from the set $\{1,\ldots,C\}$ where $C \ge 2$. The best split s for X is obtained by maximizing

$$G(s,t) = \sum_{c=1}^{C} \left[\frac{1}{n_L}\left(\sum_{i\in t_L} Z_{i(c)}\right)^2 + \frac{1}{n_R}\left(\sum_{i\in t_R} Z_{i(c)}\right)^2 \right]$$

where $Z_{i(c)} = 1_{\{Y_i = c\}}$. Now consider the multivariate classification scenario, where each outcome coordinate $Y_{i,j}$ for $q+1 \le j \le p$ is a class label from $\{1,\ldots,C_j\}$ for $C_j \ge 2$. We apply the Gini split-statistic to each coordinate yielding the extended Gini splitting rule ($r = p - q$)

(3)

$$G_r^*(s,t) = \sum_{j=q+1}^{p} \left[\frac{1}{C_j} \sum_{c=1}^{C_j} \left\{ \frac{1}{n_L}\left(\sum_{i\in t_L} Z_{i(c),j}\right)^2 + \frac{1}{n_R}\left(\sum_{i\in t_R} Z_{i(c),j}\right)^2 \right\} \right]$$

where $Z_{i(c),j} = 1_{\{Y_{i,j} = c\}}$. Note that the normalization $1/C_j$ employed for a coordinate j is required to standardize the contribution of the Gini split from that coordinate.

Multivariate normalized composite splitting rule

Observe that (2) and (3) are equivalent optimization problems, with optimization over $Y_{i,j}^*$ for regression and $Z_{i(c),j}$ for classification. This leads to similar theoretical splitting properties in regression and classification settings [2]. Given this equivalence, we can combine the two splitting rules to form a composite splitting rule. The mixed outcome splitting rule $\Theta(s,t)$ is a composite standardized split rule of mean-squared error (2) and Gini index splitting (3); i.e.,

$$\Theta(s,t) = D_q^*(s,t) + G_r^*(s,t).$$

The best split for X is the value of s maximizing $\Theta(s,t)$.

Unsupervised splitting

In usupervised mode, there is no response outcome and only feature data. Therefore in order to split the tree, the features take turns acting as the Y-outcome and X-variables. Specifically, `mtry` X-variables are randomly selected for splitting the node. Then for each of these randomly selected features, `ytry` variables are selected from the remaining features to act as the target pseduo-outcomes. Splitting uses the multivariate normalized composite splitting rule just described. As illustration, the following equivalent unsupervised calls set `mtry` to 10 and `ytry` to 5:

```
rfsrc(data = my.data, ytry = 5, mtry = 10)
```

```
rfsrc(Unsupervised(5) ~ ., my.data, mtry = 10)
```

Mahalanobis distance

As can be seen, the multivariate regression splitting rule is a composite mean-squared error rule. As such, since it is additive in the components of the outcomes, it does not take into account correlation between the continuous coordinates $(Y_1, \ldots, Y_q)^T$ of the outcome vector $\mathbf{Y} = (Y_1, \ldots, Y_p)^T$. But doing so could potentially improve performance of the forest and yield important insight into the relationship between the outcomes and the features.

In order to incorporate this potentially useful correlation, we use the Mahalanobis distance. Let $\mathbf{Z}$ be a random element with mean $\mu_{\mathbf{Z}}$ and variance-covariance $\mathbf{\Sigma}_{\mathbf{Z}}$. The Mahalanobis distance from $\mathbf{Z}$ to its mean value $\mu_{\mathbf{Z}}$ is defined by

$$\mathcal{D}_M(\mathbf{Z}) = (\mathbf{Z} - \mu_{\mathbf{Z}})^T \mathbf{\Sigma}_{\mathbf{Z}}^{-1} (\mathbf{Z} - \mu_{\mathbf{Z}}).$$

This is a sensible measure of distance but a a problem with using it in practice is that $\mathbf{\Sigma}_{\mathbf{Z}}$ may be singular (in which case $\mathbf{\Sigma}_{\mathbf{Z}}^{-1}$ does not exist). Such a scenario typically occurs when growing a tree as number of observations decreases exponentially fast. Small node sizes yield covariance matrices that are singular.

We resolve this by using the generalized inverse. Recall the definition of the Moore-Penrose generalized inverse of a matrix. For any matrix $\mathbf{A}_{n\times p}$ the generalized inverse of $\mathbf{A}$ is the unique matrix $\mathbf{A}^+_{p\times n}$ satisfying

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}, \;\; \mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+, \;\; (\mathbf{A}\mathbf{A}^+)^T = \mathbf{A}\mathbf{A}^+.$$

If $\mathbf{A}$ is non-singular, then $\mathbf{A}^+ = \mathbf{A}^{-1}$.

The generalized inverse of a matrix can be obtained from its singular value decomposition (SVD). Let $\mathbf{A}$ be an $n \times p$ matrix with rank $r \leq \min(n, p)$ where $n \geq p$. The SVD of $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where $\mathbf{U}_{n\times p}$ is an orthonormal matrix ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$), $\mathbf{V}_{p\times p}$ is an orthogonal matrix ($\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_p$) and $\mathbf{D}_{p\times p} = \text{diag}\{d_j\}_{j=1}^p$ is a diagonal matrix with non-negative entries. Without loss of generality, we can assume the singular values are ordered so that

$$d_1 \geq d_2 \geq \cdots \geq d_r > 0 \quad \text{and} \quad d_{r+1} = \cdots = d_p = 0.$$

Notice that $d_j > 0$ for $j = 1, \ldots, p$ if $\mathbf{A}$ has full column rank $r = p$.

Theorem 1: The generalized inverse for $\mathbf{A}_{n\times p}$ given that $n \geq p$ is

$$\mathbf{A}^+_{p\times n} = \mathbf{V}\mathbf{D}^+\mathbf{U}^T$$

where $\mathbf{D}^+$ is the generalized inverse of $\mathbf{D}$ defined as the $p \times p$ diagonal matrix with entries $1/d_k$ for $k = 1, \ldots, r$ and zero for coordinates $k = r + 1, \ldots, p$. If $p > n$ then transpose $\mathbf{A}$ so that the above result applies and transpose the resulting inverse. Thus, $\mathbf{A}^+ = ((\mathbf{A}^T)^+)^T$ when $p > n$.

We will be interested in symmetric square matrices of the form $\mathbf{Q} = \mathbf{L}^T\mathbf{L}$. The generalized inverse for $\mathbf{Q}$ is then easily obtained by Theorem 1 by setting $\mathbf{A} = \mathbf{Q}$. Since the number of rows equals the number of columns in $\mathbf{Q}$, the first part of Theorem 1 applies in this case.

Mahalanobis splitting rule

We use Theorem 1 to develop an efficient multivariate splitting rule based on Mahalanobis distance. Also we will no longer require the standardization used earlier (1). Hereafter, we we assume all coordinates of $\mathbf{Y}$ are continuous: thus $q = p$ and $\mathbf{Y} = (Y_1, \ldots, Y_p)^T \in \mathbb{R}^p$. Consider splitting a tree node t. Let $\mathbf{L}_t^*$ be the outcome matrix for t formed by "row-stacking'' the centered $\mathbf{Y}$ values in t (we use a $*$ notation to emphasize that the matrix is centered)

$$\mathbf{L}_t^* = \begin{pmatrix} (\mathbf{Y}_1 - \overline{\mathbf{Y}}_t)^T \\ \vdots \\ (\mathbf{Y}_n - \overline{\mathbf{Y}}_t)^T \end{pmatrix}_{n \times p}.$$

Here $\overline{\mathbf{Y}}_t$ is the p-dimensional vector of sample means for $\mathbf{Y}$ in t. If there are n observations in t, then $\mathbf{L}_t^*$ is an $n \times p$ matrix. Because $\mathbf{L}_t^*$ is centered, the sample covariance matrix for the data is $n^{-1}\mathbf{Q}_t^*$ where $\mathbf{Q}_t^* = (\mathbf{L}_t^*)^T\mathbf{L}_t^*$. Apply Theorem 1section to $\mathbf{Q}_t^*$ to obtain its generalized inverse $(\mathbf{Q}_t^*)^+$.

Now we define the splitting rule. Suppose that t is split into left and right daughter nodes t_L and t_R (remember splitting is based on the features, $\mathbf{X}$). Let $\overline{\mathbf{Y}}_{t_L}$ and $\overline{\mathbf{Y}}_{t_R}$ be the p-dimensional sample mean vectors for $\mathbf{Y}$ in t_L and t_R, respectively. The Mahalanobis multivariate split-statistic is

$$\begin{aligned} &\mathcal{D}_{M,t}(t_L, t_R) \\ &\quad = \frac{n_L}{n} \sum_{i \in t_L} (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_L})^T (\mathbf{Q}_t^*)^+ (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_L}) + \frac{n_R}{n} \sum_{i \in t_R} (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_R})^T (\mathbf{Q}_t^*)^+ (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_R}). \end{aligned}$$

The best split for t is obtained by minimizing $\mathcal{D}_{M,t}(t_L, t_R)$, or equivalently by maximizing

$$\mathcal{D}_{M,t}^*(t_L, t_R) = 1 - \frac{1}{p}\mathcal{D}_{M,t}(t_L, t_R).$$

This split-statistic is always bounded between 0 and 1 as the following theorem shows.

Theorem 2: $0 \le \mathcal{D}_{M,t}^*(t_L, t_R) \le 1$.

Proof: We have the following bound for $\mathcal{D}_{M,t}(t_L, t_R)$:

$$\begin{aligned} &\frac{n_L}{n} \sum_{i \in t_L} (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_L})^T (\mathbf{Q}_t^*)^+ (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_L}) + \frac{n_R}{n} \sum_{i \in t_R} (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_R})^T (\mathbf{Q}_t^*)^+ (\mathbf{Y}_i - \overline{\mathbf{Y}}_{t_R}) \\ &\quad \le \sum_{i=1}^{n} (\mathbf{Y}_i - \overline{\mathbf{Y}})^T (\mathbf{Q}_t^*)^+ (\mathbf{Y}_i - \overline{\mathbf{Y}}) \\ &\quad = \text{trace}\left\{ (\mathbf{Q}_t^*)^+ \cdot \sum_{i=1}^{n} (\mathbf{Y}_i - \overline{\mathbf{Y}})(\mathbf{Y}_i - \overline{\mathbf{Y}})^T \right\} \\ &\quad = \text{trace}\left\{ (\mathbf{Q}_t^*)^+ \mathbf{Q}_t^* \right\} \\ &\quad = \text{trace}\left\{ \mathbf{V}\mathbf{D}^+\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T \right\} \\ &\quad = \text{trace}\left\{ \mathbf{V}\mathbf{D}^+\mathbf{D}\mathbf{V}^T \right\} \\ &\quad \le \text{trace}\left\{ \mathbf{V}^T\mathbf{V} \right\} = p, \end{aligned}$$

where line 3 follows from $\sum_{i=1}^n \mathbf{z}_i^T \mathbf{A} \mathbf{z}_i = \text{trace}(\mathbf{AB})$ where $\mathbf{B} = \sum_{i=1}^n \mathbf{z}_i \mathbf{z}_i^T$. Therefore

$$1 \ge 1 - \frac{1}{p}\mathcal{D}_{M,t}(t_L, t_R) \ge 1 - \frac{1}{p} \cdot p \ge 0.$$

Illustration

We use a mouse nutrigenomic study [3] to illustrate variable selection using Mahalanobis splitting and constrast the results to the default composite splitting rule. The **Y**-outcome is a 22-dimensional vector of liver lipids (all real-valued) which is regressed against features **X** comprising 120 gene expression value, mouse genotype (wild-type or PPAR-alpha) and type of diet (5 different treatments). Comparison of the standardized variable importance (VIMP) of the two splitting rules, shows that the independence rule tends to favor the variables genotype and diet, whereas Mahalanobis splitting is able to find not only these variables, but also some interesting genes, such as CYP3A11, with large importance.

```
## ------------------------------------------------------------
## multivariate regression forests
## - comparison of Mahalanobis splitting to standard splitting rule
## - lipids (all real values) used as the multivariate y
## - genes, genotype and diet used as the x features
## ------------------------------------------------------------
library(randomForestSRC)
## load the data
data(nutrigenomic)

## parse into y and x data
ydta <- nutrigenomic$lipids
xdta <- data.frame(nutrigenomic$genes,
                   diet = nutrigenomic$diet,
                   genotype = nutrigenomic$genotype)

## mahalanobis splitting
obj <- rfsrc(get.mv.formula(colnames(ydta)),
             data.frame(ydta, xdta),
             importance=TRUE, nsplit = 10, splitrule = "mahalanobis")

## default composite (independence) splitting
obj2 <- rfsrc(get.mv.formula(colnames(ydta)),
              data.frame(ydta, xdta),
              importance=TRUE, nsplit = 10)

## compare standardized VIMP for top 25 variables
imp <- data.frame(mahalanobis = rowMeans(get.mv.vimp(obj,  standardize = TRUE)),
                  default     = rowMeans(get.mv.vimp(obj2, standardize = TRUE)))
print(100 * imp[order(imp["mahalanobis"], decreasing = TRUE)[1:25], ])

>          mahalanobis      default
> diet       4.6368819 19.674080838
> CYP3A11    2.7681390  2.540102290
> PMDCI      1.8182919  2.815791465
> genotype   1.5269731  3.553100850
> CYP2c29    1.4269237  0.531133113
> Ntcp       1.0038941  0.484867059
> CAR1       0.7721098  1.022618362
> CYP4A10    0.7699449  0.073722892
> GSTpi2     0.7569144  0.076668861
```

```
> ACAT2      0.5973348  0.466176034
> SPI1.1     0.5583821  0.520315157
> THIOL      0.4217010  0.256322710
> G6Pase     0.3452151  0.076882606
> L.FABP     0.3264878  0.202130329
> SR.BI      0.2879030  0.595221688
> BIEN       0.2873616  0.173956271
> FAS        0.2752811 -0.068033669
> ACOTH      0.2733822  0.393634169
> apoC3      0.2591111  0.921950081
> CYP4A14    0.2315877  0.105256113
> GSTmu      0.2165187  0.062434129
> Lpin1      0.1948898  0.327322317
> LDLr       0.1921990  0.107747940
> ACBP       0.1895075  0.696519223
> UCP2       0.1850553 -0.005560895
```

Cite this vignette as

H. Ishwaran, F. Tang, M. Lu, and U. B. Kogalur. 2021. "randomForestSRC: multivariate splitting rule vignette." http://randomforestsrc.org/articles/mvsplit.html.

```
@misc{HemantMultiv,
    author = "Hemant Ishwaran and Fei Tang and Min Lu and Udaya B. Kogalur",
    title = {{randomForestSRC}: multivariate splitting rule vignette},
    year = {2021},
    url = {http://randomforestsrc.org/articles/mvsplit.html}
}
```

References

1. Tang F, Ishwaran H. Random forest missing data algorithms. Statistical Analysis and Data Mining. 2017;10:363–77.
2. Ishwaran H. The effect of splitting on random forests. Machine Learning. 2015;99:75–118.
3. Martin PG, Guillou H, Lasserre F, Déjean S, Lan A, Pascussi J-M, et al. Novel aspects of PPARα-mediated regulation of lipid and xenobiotic metabolism revealed through a nutrigenomic study. Hepatology. 2007;45:767–77.