Library missForest

Eunice García Chambi

Concepto

El paquete missForest (Non parametric Missing Value imputation using Random Forest) **permite imputar valores perdidos para el caso de datos de tipo mixto**, es decir imputa datos continuos y/o categóricos a través de interacciones complejas y relaciones no lineales.

Datos de aplicación del paquete missForest

En la aplicación de las funciones del paquete missForest, utilizaremos el conjunto de datos iris. Las variables o atributos que se miden de de cada flor son del siguiente tipo:

1. El tipo de flor como variable categórica.
2. El largo y el ancho del pétalo en cm. como variables numéricas.
3. El largo y el ancho del sépalo en cm. como variables numéricas.

A continuación se ofrece una pequeña visualización de las primeras cinco observaciones del conjunto de datos:

```
head(iris, n=5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

Funciones

1. **varClass** Devuelve los tipos de variables de un marco de datos. Se usa internamente en varias funciones del paquete 'missForest'.

```
library(missForest)
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## Loading required package: foreach
```

```
## Loading required package: itertools
```

```
## Loading required package: iterators
```

```
varClass(iris)
```

```
## [1] "numeric" "numeric" "numeric" "numeric" "factor"
```

2 prodNA

Introduce artificialmente valores perdidos. Las entradas en el marco de datos dado se eliminan completamente al azar hasta la cantidad especificada.

```
library(missForest)
prodNA(iris, noNA = 0.2)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1          NA          1.4          NA     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2           NA         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6          NA          1.4         0.3     setosa
## 8            5.0          NA          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12            NA          NA          1.6         0.2     setosa
## 13            NA         3.0          1.4         0.1       <NA>
## 14           4.3          NA          1.1         0.1       <NA>
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17            NA         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7          NA     setosa
## 20           5.1         3.8          1.5         0.3       <NA>
```

3 mixError

se usa para calcular el error de imputación, particularmente en el caso de datos de tipo mixto. Dada la matriz de datos completa y la matriz de datos que contiene los valores faltantes, se calcula el error cuadrático medio normalizado raíz para el continuo y la proporción de entradas clasificadas falsamente para las variables categóricas.

```
mixError(ximp, xmis, xtrue)
```

Argumentos

- **ximp:** matriz de datos imputada con variables en las columnas y observaciones en las filas. Tenga en cuenta que no debe faltar ningún valor.
- **xmis:** matriz de datos con valores faltantes.
- **xtrue:** Matriz de datos completa. Tenga en cuenta que no debe faltar ningún valor.

4. **missForest**

Argumentos

- **xmis**: una matriz de datos con valores faltantes.
- **maxiter**: el número máximo de iteracciones a realizar.
- **variablewise**: lógico. Si es "TRUE" se devuelve el error OOB para cada variable por separado. Esto puede ser útil como control de fiabilidad para las variables imputadas para un subsiguiente análisis de datos.
- **decreasing**: lógico. Si es "FALSE"entonces las variables son creadas en orden creciente de entradas perdidas.
- **verbose**: lógico. Si es "TRUE"el usuario recibe una salida adicional entre iteracciones, con el error de imputación estimado a través del tiempo de ejecución y si se suministra la matriz completa de datos se recibe el verdadero error de imputación.

-**Iris data**:Este conjunto de datos completo contiene cinco variables, una de las cuales es categórica con tres niveles. Está contenido en la base R y se puede cargar directamente escribiendo data(iris). Los datos fueron recopilados por Anderson [1935].

Ejemplos:

```
data(iris)
iris.mis <- prodNA(iris, noNA = 0.1)
summary(iris.mis)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.400   Median :1.300
##  Mean   :5.834   Mean   :3.068   Mean   :3.804   Mean   :1.172
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##  NA's   :16      NA's   :12      NA's   :13      NA's   :15
##        Species
##  setosa    :45
##  versicolor:44
##  virginica :42
##  NA's      :19
##
##
##
```

```
iris.imp <- missForest(iris.mis)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
##   missForest iteration 6 in progress...done!
```

-Podemos recurrir a la matriz de datos imputados escribiendo iris.imp$ximp.

-**Nota**: Un error común es usar iris.imp en lugar de iris.imp$ximp para análisis posteriores

-Además, missForest proporciona una estimación de error de imputación OOB que se puede extraer utilizando la misma notación $ que con la matriz de datos imputada:

```
iris.imp$OOBerror
```

```
##      NRMSE        PFC
## 0.14094606 0.05343511
```

```
iris.imp <- missForest(iris.mis, variablewise = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
```

```
 iris.imp$OOBerror
```

```
##        MSE        MSE        MSE        MSE        PFC
## 0.11671756 0.10471926 0.06879029 0.03403059 0.04580153
```

```r
set.seed(81)
iris.imp <- missForest(iris.mis, verbose = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##     estimated error(s): 0.1599052 0.06870229
##     difference(s): 0.006187461 0.09333333
##     time: 0.095 seconds
##
##   missForest iteration 2 in progress...done!
##     estimated error(s): 0.1429516 0.06870229
##     difference(s): 0.0001182103 0
##     time: 0.097 seconds
##
##   missForest iteration 3 in progress...done!
##     estimated error(s): 0.1412779 0.05343511
##     difference(s): 5.348265e-05 0
##     time: 0.084 seconds
##
##   missForest iteration 4 in progress...done!
##     estimated error(s): 0.1402429 0.04580153
##     difference(s): 4.90784e-05 0
##     time: 0.094 seconds
##
##   missForest iteration 5 in progress...done!
##     estimated error(s): 0.1392263 0.0610687
##     difference(s): 1.116175e-05 0
##     time: 0.091 seconds
##
```

```
iris.imp$OOBerror
```

```
##     NRMSE       PFC
## 0.1392263 0.0610687
```

- **ntree**: El efecto de reducir ntree en el tiempo de cálculo es lineal, por ejemplo, reducir a la mitad ntree será la mitad Tiempo de cálculo para una sola iteración. El valor predeterminado en missForest se establece en 100, que es realmente largo.

-Ejemplo

```
iris.mis <- prodNA(iris, 0.05)
iris.imp <- missForest(iris.mis, verbose = TRUE, maxiter = 3)
```

```
##   missForest iteration 1 in progress...done!
##     estimated error(s): 0.1495225 0.05594406
##     difference(s): 0.002065166 0.04
##     time: 0.094 seconds
##
##   missForest iteration 2 in progress...done!
##     estimated error(s): 0.1392422 0.06293706
##     difference(s): 1.018861e-05 0
##     time: 0.098 seconds
##
##   missForest iteration 3 in progress...done!
##     estimated error(s): 0.141979 0.06293706
##     difference(s): 7.800306e-06 0
##     time: 0.098 seconds
```

```
iris.imp <- missForest(iris.mis, verbose = TRUE, maxiter = 3, ntree = 20)
```

```
##   missForest iteration 1 in progress...done!
##     estimated error(s): 0.1664045 0.07692308
##     difference(s): 0.002098089 0.04
##     time: 0.028 seconds
##
##   missForest iteration 2 in progress...done!
##     estimated error(s): 0.1468648 0.04895105
##     difference(s): 3.177022e-05 0
##     time: 0.028 seconds
##
##   missForest iteration 3 in progress...done!
##     estimated error(s): 0.1462276 0.06293706
##     difference(s): 4.499018e-05 0
##     time: 0.024 seconds
```

- **replace**: lógico. Si es “TRUE”se utiliza muestreo bootstrap (con reemplazos).
- **classwt**: lista de probabilidades a priori de las clases en las variables categóricas. Esto es equivalente al argumento randomForest, sin embargo en este caso el usuario tiene que establecerla para todas las variables en el conjunto de datos (para las variables continuas se establece “NULL”).
- **cutoff** : lista de puntos de corte de clase para cada variable categórica. Igual que con el argumento classwt para las variables continuas se establece en “1”.
- **strata**: lista de variables (factor) utilizadas para el muestreo estratificado. Igual que con el argumento classwt para las variables continuas se establece en “NULL”.
- **sampsize**: lista de tamaño(s) de muestra. Esto es equivalente al argumento randomForest, sin embargo, en este caso el usuario tiene que establecer los tamaños para todas las variables.

- **nodesize**: tamaño mínimo de los nodos terminales. Tiene que ser un vector de longitud 2, donde el primero representa la entrada del número de las variables continuas y el segundo el número de las variables categóricas. El valor predeterminado es 1 para continuas y 5 para variables categóricas.
- **maxnodes**: número máximo de nodos terminales.
- **xtrue**: opcional a través de la matriz completa de datos. Esto se puede aplicar para probar el rendimiento. Al proporcionar la matriz de datos completa el argumento verbose mostrará la verdadera imputación después de cada iteración y la salida también contendrá el error de imputación.

-Ejemplo

```
iris.imp <- missForest(iris.mis, xtrue = iris, verbose = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##     error(s): 0.1520561 0
##     estimated error(s): 0.1510632 0.06993007
##     difference(s): 0.00206499 0.04
##     time: 0.099 seconds
##
##   missForest iteration 2 in progress...done!
##     error(s): 0.150827 0
##     estimated error(s): 0.1405473 0.04895105
##     difference(s): 9.658662e-06 0
##     time: 0.094 seconds
##
##   missForest iteration 3 in progress...done!
##     error(s): 0.147658 0
##     estimated error(s): 0.1403714 0.06993007
##     difference(s): 4.714667e-06 0
##     time: 0.098 seconds
##
##   missForest iteration 4 in progress...done!
##     error(s): 0.1472411 0
##     estimated error(s): 0.1410152 0.06293706
##     difference(s): 5.355424e-06 0
##     time: 0.099 seconds
```

- **parallelize**: indica si la función missForest() debe ejecutarse en paralelo. El valor predeterminado es “no”. Si se utiliza “variables” los datos son divididos en trazos de tamaño igual al número de núcleos registrados en el paralelo backend (backend hace referencia al conjunto de aplicaciones que gestionan el proceso y almacenamiento de los datos en la aplicación de la función missForest()) . Si se utiliza “forests”el número total de árboles aleatorios se divide de la misma manera.

```r
library(missForest)
missForest(xmis=iris, maxiter = 10, ntree = 100, variablewise = FALSE,
           decreasing = FALSE, verbose = FALSE,
           mtry = floor(sqrt(ncol(xmis))), replace = TRUE,
           classwt = NULL, cutoff = NULL, strata = NULL,
           sampsize = NULL, nodesize = NULL, maxnodes = NULL,
           xtrue = NA, parallelize = c('no', 'variables', 'forests'))
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
```

```
## $ximp
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
```

```r
library(missForest)
## Nonparametric missing value imputation on mixed-type data:
data(iris)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```r
## The data contains four continuous and one categorical variable.

## Artificially produce missing values using the 'prodNA' function:
set.seed(81)
iris.mis <- prodNA(iris, noNA = 0.2)
summary(iris.mis)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
```