



FSinR: Un paquete extenso para selección de características

Francisco Aragón¹, Alfonso Jiménez², Antonio Arauzo², José M. Benítez¹

¹ *Depto. Ciencias de la Computación e Inteligencia Artificial, DECSAI, DICITS*

Universidad de Granada

² *Área de Proyectos de Ingeniería*

Universidad de Córdoba



XI Jornadas de Usuarios de R

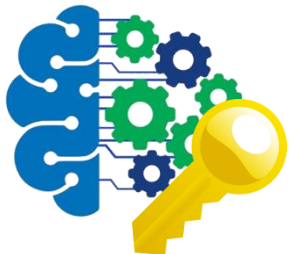
INTRODUCCIÓN

¿Qué es selección de características?

- Es el proceso de seleccionar un subconjunto de características relevantes



- Se ha convertido en parte clave del Machine Learning



Feature Selection

¿Por qué es importante?

- Su aplicación presenta un impacto directo en el rendimiento de los modelos
 - Elimina información redundante, irrelevante, ruido, ...
 - Facilita la interpretación
 - Reduce el tiempo de cómputo
 - Reduce la complejidad del modelo
 - ...
- Mejora los resultados de los modelos



MOTIVACIÓN

Es un problema vigente

- La Selección de Características es un proceso bastante complejo
- Esto hace que aún no se haya encontrado una solución definitiva



- Por lo tanto, siguen siendo interesantes las nuevas propuestas y aportaciones

¿Qué hay en R sobre FS?

- Paquetes dedicados con varios métodos:
 - FSelector
 - MXM
- Paquetes dedicados a un método:
 - Boruta
 - spSRF
 - varSelRF
 - ...
- Paquetes no dedicados:
 - caret
 - CORElearn
 - ...



¿Qué vemos que falta?

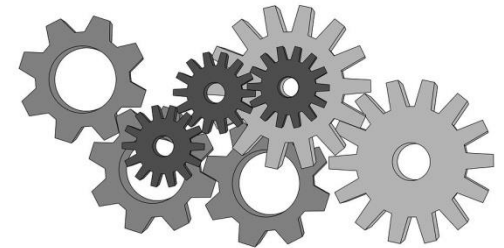
- Los paquete actuales abordan el problema de una forma poco extensa:
 - Tienen pocos métodos...
 - ... ó solo un método concreto
 - ... ó no están ni siquiera dedicados
- No contienen una gran variedad de métodos, ni un gran número de métodos conocidos en el estado del arte, ni dan facilidades de uso...
- Falta un gran paquete dedicado a la Selección de Características

PRINCIPAL PROBLEMA



¿Qué aporta nuestro paquete?

- Gran recopilación de métodos de filtro y wrapper muy usados en la literatura
 - + de 15 métodos de filtro
 - Para los métodos de wrapper se usa el paquete caret
 - + de 200 modelos de regresión y clasificación
- Estrategias de búsqueda
 - Combinación con filtro y wrapper
- Fácil de usar, amplia documentación, vignettes de uso, web con más detalle...

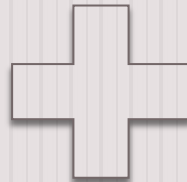


PAQUETE FSinR

Paquete FSinR

Es un paquete R con herramientas
para realizar el
Proceso de Selección de Características

Algoritmo de búsqueda



Método de Wrapper

Método de Filtro

Métodos de Wrapper

Métodos de Filtro

Métodos de
corte

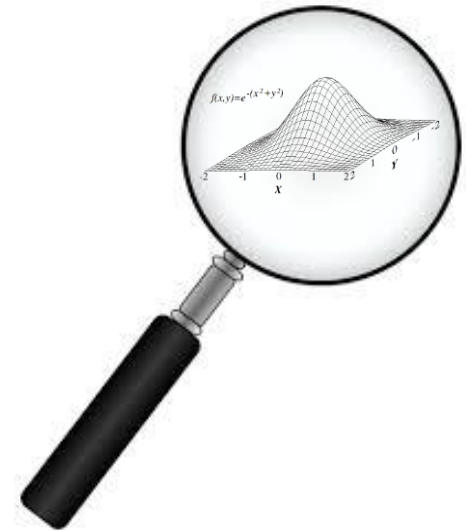


Métodos de
Wrapper

Métodos de Filtro

Métodos de búsqueda

- Guían la búsqueda de la mejor solución en el espacio formado por todas las combinaciones de características
 - Secuenciales (SFS, SFFS, SBS, SFBS)
 - Anchura / profundidad
 - Local (Hill-Climbing, Taboo Search)
 - Probabilísticas (Las Vegas)
 - Heurísticas (GA, SA, WOA, ACO)
- Se aplica con métodos de filtro y wrapper



Métodos de filtro

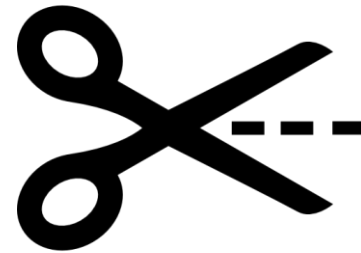
- Devuelven la valoración de un subconjunto de características basándose en test estadísticos
- Individuales
 - Chi-Squared, Cramer V, F-score, Relief
- Conjunto
 - Rough Sets Consistency, Binary Consistency, IE Consistency, IEP Consistency, Determination Coefficient, Mutual Information, Gain Ratio, Symmetrical Uncertain, Gini Index, Jd Evaluation, MDLC Evaluation, RFSM Evaluation
- Híbridas
 - LCC

Métodos de wrapper

- Devuelven una evaluación de un subconjunto de características por medio de un algoritmo de aprendizaje
- Se puede hacer uso de los 238 modelos disponibles del paquete caret
- Por medio de la función *wrapperGenerator*, que define:
 - Los parámetros de remuestreo (*trainControl* de caret)
 - Los parámetros de ajuste (*train* de caret)
 - Nombre del método de aprendizaje (*de los disponibles en caret*)

Métodos de corte

- Las medidas de corte eligen un subconjunto de características según una serie de criterios
 - Select k-best
 - Select percentile
 - Select Threshold
 - Select Threshold range
 - Select difference
 - Select Slope
- Se pueden usar con métodos de filtro y de wrapper (evaluación individual)



EJEMPLOS DE USO

Búsqueda con wrapper (clas.)

```
1 library(FSInR)
2
3 # Values for trainControl function
4 resamplingParams <- list(method = "cv", number = 10)
5 # Values for train function (x, y, method and trainControl not necessary)
6 fittingParams <- list(preProc = c("center", "scale"), metric="Accuracy", tuneGrid = expand.grid(k = c(1:20)))
7
8 # wrapper method
9 wrapper <- wrapperGenerator("knn", resamplingParams, fittingParams)
10
11 # search method (sfs) + wrapper method
12 sfs(iris, 'Species', wrapper)
13 # search method (ga) + wrapper method
14 ga(iris, 'Species', wrapper, popSize = 8, pcrossover = 0.8, pmutation = 0.1, maxiter=4, verbose=TRUE)
15 # search method (hc) + wrapper method
16 hc(iris, 'Species', wrapper, verbose=TRUE)
17
18 # ...
```

Salida por consola

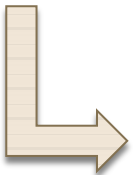


HC		InitialVector = 0010		InitialFitness = 0.9533333
HC		Vector = 0011		Fitness = 0.9666667 End? = no
HC		Vector = 1011		Fitness = 0.9733333 End? = no
HC		Vector = 0011		Fitness = 0.9666667 End? = yes

Búsqueda con wrapper (reg.)

```
1 library(FSInR)
2
3 # Values for trainControl function
4 resamplingParams <- list(method = "repeatedcv", repeats = 3)
5 # Values for train function (x, y, method and trainControl not necessary)
6 fittingParams <- list(preProc = c("center", "scale"), metric="RMSE",
7                       tuneGrid = expand.grid(size = seq(1,12,by=2), decay=0), trace=FALSE)
8
9 # wrapper method
10 wrapper <- wrapperGenerator("nnet",resamplingParams, fittingParams)
11
12 # search method (sbs) + wrapper method
13 sbs(mtcars, 'mpg', wrapper)
14 # search method (sa) + wrapper method
15 sa(mtcars, 'mpg', wrapper, temperature = 5, temperature_min=0.01, reduction=0.6, innerIter=1, verbose=TRUE)
16 # search method (ts) + wrapper method
17 ts(mtcars, 'mpg', wrapper, numNeigh = 4, tamTabuList = 4, iter = 5, intensification=1, iterIntensification=5,
18    diversification=1, iterDiversification=5, verbose=TRUE)
19
20 # ...
```

Salida por consola




TS		InitialVector = 0011011111		InitialFitness = 19.7355644	
TS		Iter = 1		Vector = 0011010111	Fitness = 19.6899545 BestFitness = 19.6899545
TS		Iter = 2		Vector = 0011010011	Fitness = 19.6207992 BestFitness = 19.6207992
TS		Iter = 3		Vector = 1011010011	Fitness = 19.6620354 BestFitness = 19.6207992
TS		Iter = 4		Vector = 1011000011	Fitness = 19.6026519 BestFitness = 19.6026519
TS		Iter = 5		Vector = 1011001011	Fitness = 19.4186876 BestFitness = 19.4186876

Búsqueda con filtro (clas.)

```
1 library(FSInR)
2
3 # search method (sffs) + filter method
4 sffs(iris, 'Species', giniIndex)
5
6 # ...
```

Resultados



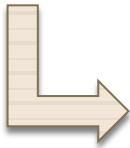
```
$bestFeatures
      Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]              1          1           1           0

$bestValue
[1] 1
```

Filtro / Wrapper individual

```
1 library(FSInR)
2
3 chiSquared(iris, 'Species', c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"))
```

Resultados



Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
156.26667	89.54629	271.80000	271.75000

```
1 library(FSInR)
2
3 # Values for trainControl function
4 resamplingParams <- list(method = "cv", number = 10)
5 # Values for train function (x, y, method and trainControl not necessary)
6 fittingParams <- list(preProc = c("center", "scale"), metric="Accuracy", tuneGrid = expand.grid(k = c(1:20)))
7
8 # wrapper method
9 wrapper <- wrapperGenerator("knn", resamplingParams, fittingParams)
10
11 wrapper(iris, "Species", c("Sepal.Length", "Sepal.Width"))
```

Resultados



[1] 0.8133333

Métodos de corte

```
1 library(FSInR)
2
3 # Values for trainControl function
4 resamplingParams <- list(method = "cv", number = 10)
5 # Values for train function
6 fittingParams <- list(preProc = c("center", "scale"), metric="Accuracy", tuneGrid = expand.grid(k = c(1:20)))
7
8 # wrapper method
9 wrapper <- wrapperGenerator("knn", resamplingParams, fittingParams)
10
11 # cutoff with wrapper
12 selectKBest(iris, 'Species', wrapper, 2)
13
14 # cutoff with filter
15 selectThreshold(iris, 'Species', cramer, 0.70)
```

Resultados

```
$bestFeatures
  Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]          0          0           1           1

$featuresSelected
[1] "Petal.Width" "Petal.Length"

$valuePerFeature
[1] 0.9600000 0.9533333
```

Resultados

```
$bestFeatures
  Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]          1          0           1           1

$featuresSelected
[1] "Petal.Length" "Petal.Width" "Sepal.Length"

$valuePerFeature
[1] 0.9518403 0.9517528 0.7217263
```

CONCLUSIÓN

Conclusiones

- La Selección de Características es importante dentro del Machine Learning
- Existe una carencia de paquetes extensos dedicados en R
- En este sentido hemos presentado el paquete FSinR que destaca por:
 - Métodos de filtro y wrapper más comunes en la literatura
 - Combinación con métodos de búsqueda y corte
 - La herramienta más extensa y completa
 - Fácil de usar, muy documentada

FSinR



GRACIAS