

fcaR

Un paquete R para manipulación de implicaciones difusas

D. López y A. Mora

XI Jornadas de Usuarios de R



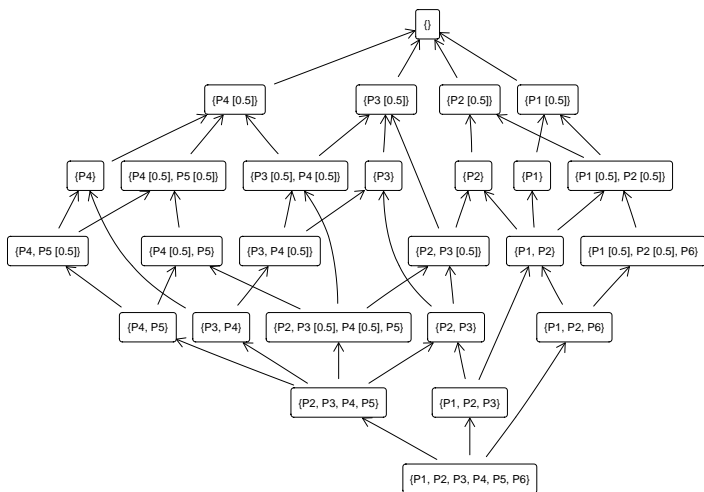
MATEMÁTICA APLICADA
UNIVERSIDAD DE MÁLAGA

Análisis de Conceptos Formales

	P1	P2	P3	P4	P5	P6
O1	0.0	1.0	0.5	0.5	1.0	0
O2	1.0	1.0	1.0	0.0	0.0	0
O3	0.5	0.5	0.0	0.0	0.0	1
O4	0.0	0.0	0.0	1.0	0.5	0
O5	0.0	0.0	1.0	0.5	0.0	0
O6	0.5	0.0	0.0	0.0	0.0	0

- Conceptos: conjuntos cerrados (atributos que aparecen juntos siempre)

```
fc_I$plot_lattice(object_names = FALSE)
```



- Implicaciones:

```
fc_I$implications$get_rules(3)$to_latex()
```

$$\{^{0.5}/P3, ^{0.5}/P4, ^{0.5}/P5\} \Rightarrow \{P2, P5\}$$

Paquetes de R para Implicaciones

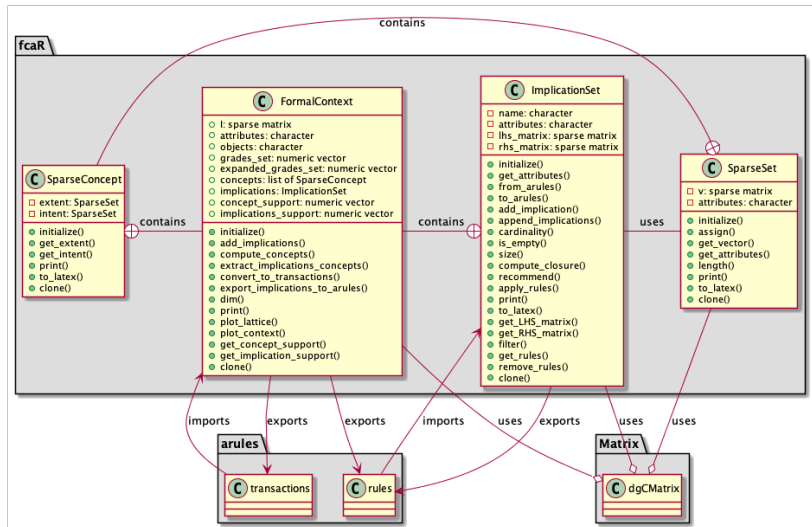
- **arules**: Proporciona la infraestructura para representar, manipular y analizar datos de transacciones y sus patrones (frequent itemsets y reglas de asociación). También proporciona implementación en C de algoritmos de minería de reglas de asociación, como Apriori y Eclat.
- **frbs**: Implementación de varios algoritmos de aprendizaje basados en sistemas de reglas difusas para tareas de clasificación y regresión.
- **RKEEL**: interfaz en R para KEEL, una popular librería en Java para tareas de extracción de conocimiento.

El objetivo principal es crear un paquete para R capaz de:

- Manipular contextos formales y encontrar conceptos.
- Extraer implicaciones a partir de un contexto.
- Proporcionar herramientas para visualizar el conocimiento extraído.
- Calcular recomendaciones (y cierres).
- Integrarse con arules.

- `FormalContext` encapsula la definición de un contexto formal (G, M, I) , siendo G el conjunto de objetos, M el de atributos e I la matriz de relación (difusa). Proporciona métodos para operar en el contexto usando herramientas de FCA.
- `ImplicationSet` representa un conjunto de implicaciones definidas sobre los atributos de un contexto formal.
- `SparseSet` es una clase para facilitar la entrada de datos y su visualización, encapsulando en formato disperso un conjunto difuso.
- `SparseConcept` encapsula internamente las dos partes de un concepto formal, el *extent* y el *intent*.

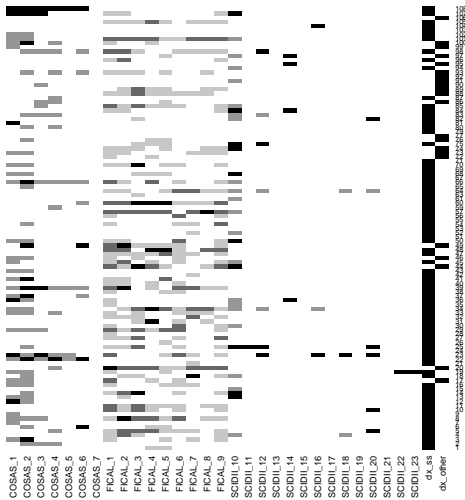
Diagrama UML



Ejemplo: Sistema Recomendador Diagnóstico

El dataset `cobre32` consiste en 30 atributos difusos que se relacionan con distintas características de trastornos relacionados con la esquizofrenia y desórdenes esquizoafectivos, incluyendo trastorno bipolar o depresión mayor, medidos sobre 105 pacientes. 2 atributos (`dx_ss` y `dx_other`) indican si cada paciente tenía esquizofrenia estricta u otro trastorno.

Los síntomas son multivaluados: sus grados van desde *ausente* hasta *extremo*, pasando por *mínimo*, *leve*, *moderado*, *moderado grave* y *grave*, que se mapean a valores en $[0, 1]$.



Inicializamos el contexto formal y extraemos las implicaciones y los conceptos relacionados.

```
library(fcaR)

# Creamos el contexto formal
fc <- formal_context$new(cobre32)

# Extraemos las implicaciones
fc$extract_implications_concepts()
```

```
# Simplificamos las reglas  
fc$implications$apply_rules(c("composition",  
                             "generalization",  
                             "simplification"))
```

```
## Using parallel execution  
## Processing batch  
## --> composition : from 985 to 985 in 0.007 secs.  
## --> generalization : from 985 to 985 in 0.039 secs.  
## --> simplification : from 985 to 985 in 3.757 secs.  
## Batch took 3.805 secs.
```

```
# Eliminamos aquellas que no tienen soporte  
soporte <- fc$get_implication_support()  
fc$implications$remove_rules(which(soporte == 0))
```

Podemos usar las implicaciones para generar recomendaciones de diagnóstico:

```
diagnose <- function(S) {  
  
  fc$implications$recommend(S = S,  
                             attribute_filter = c("dx_ss",  
                                                  "dx_other"))  
  
}
```

Generar recomendaciones:

```
S1 <- sparse_set$new(attributes = fc$attributes)
S1$assign(attributes = c("COSAS_1", "COSAS_2", "COSAS_3", "COSAS_4",
                        "COSAS_5", "COSAS_6"),
          values = c(0.5, 1, 0.5, 0.166667, 0.5, 1))

diagnose(S1) # Schizophrenia strict
```

```
##      dx_ss dx_other
##          1         0
```

```
S2 <- sparse_set$new(attributes = fc$attributes)
S2$assign(attributes = c("FICAL_1", "FICAL_2", "COSAS_1"),
          values = c(0, 1, 0))

diagnose(S2) # Not enough information
```

```
##      dx_ss dx_other
##          0         0
```

```
S3 <- sparse_set$new(attributes = fc$attributes)
S3$assign(attributes = c("COSAS_4", "FICAL_3", "FICAL_5", "FICAL_8"),
          values = c(0.666667, 0.5, 0.5, 0.5))

diagnose(S3) # Other, not schizophrenia strict
```

```
##      dx_ss dx_other
##          0         1
```

Cuando no disponemos de información, la lógica subyacente puede proporcionar información:

```
cl <- fc$implications$compute_closure(S2, reduce = TRUE)
```

```
# Escribimos las primeras reglas
```

```
cl$implications$filter(rhs = c("dx_ss", "dx_other"),  
                      drop = TRUE)$get_rules(1:5)
```

```
## Implication set with 5 implications.
```

```
## Rule 1: {SCIDII_11 [0.333333333333333]} -> {dx_ss}
```

```
## Rule 2: {FICAL_9 [0.333333333333333], SCIDII_14} -> {dx_ss}
```

```
## Rule 3: {SCIDII_14} -> {dx_ss}
```

```
## Rule 4: {COSAS_6 [0.5], SCIDII_10 [0.5]} -> {dx_ss}
```

```
## Rule 5: {COSAS_6 [0.5], FICAL_7 [0.333333333333333]} -> {dx_ss}
```

Se puede:

- Importar/Exportar transactions desde arules.
- Importar reglas de asociación, manipularlas (simplificarlas, eliminando redundancias, y todo lo que se ha visto antes, que no se puede hacer desde arules) y después devolverlas al formato de arules, incluso calculando parámetros de calidad de las mismas.

Hemos presentado un paquete para R capaz de:

- Manipular contextos formales y encontrar conceptos.
- Extraer implicaciones a partir de un contexto.
- Proporcionar herramientas para visualizar el conocimiento extraído.
- Calcular recomendaciones (y cierres).
- Integrarse con arules.

¡Muchas gracias!