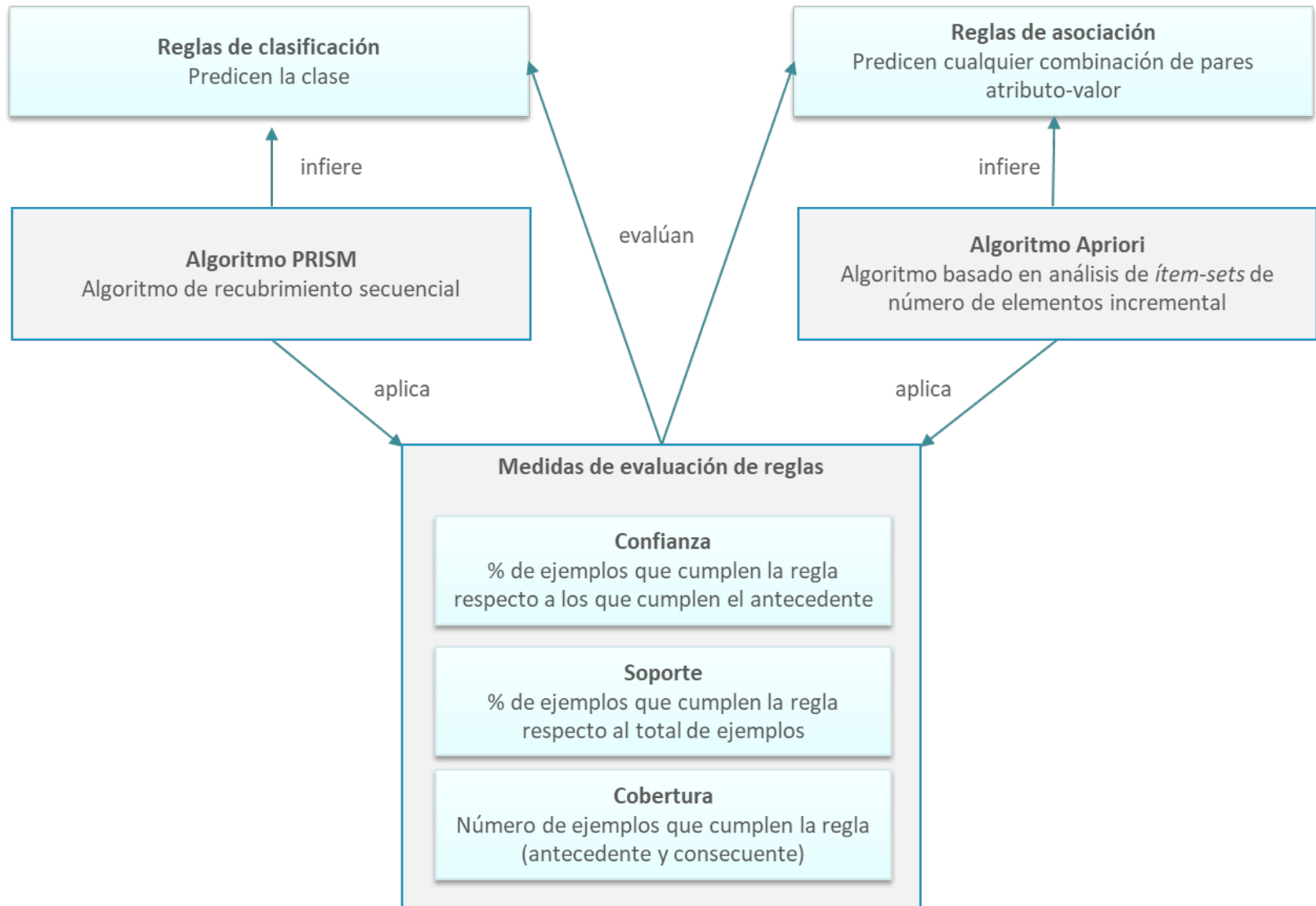


Tema 4. Reglas

Contenidos

- ▶ [4.1] Esquema y objetivos
- ▶ [4.2] Reglas de clasificación y reglas de asociación
- ▶ [4.3] Algoritmos de aprendizaje de reglas de clasificación
- ▶ [4.4] Algoritmos de aprendizaje de reglas de asociación
- ▶ [4.5] Aplicaciones y ejemplos de implementación

Reglas



Objetivos

- ▶ Representar conocimiento mediante reglas de clasificación y de asociación
- ▶ Aplicar algoritmos básicos de construcción de reglas para resolver problemas de aprendizaje
- ▶ Identificar aplicaciones prácticas de las técnicas de aprendizaje de reglas de clasificación o asociación

Representación del conocimiento mediante reglas

- ▶ Partes de una regla:
 - **Antecedente:** condición
 - **Consecuente:** conclusión o acción

```
SI <antecedente>  
    ENTONCES <consecuente>
```

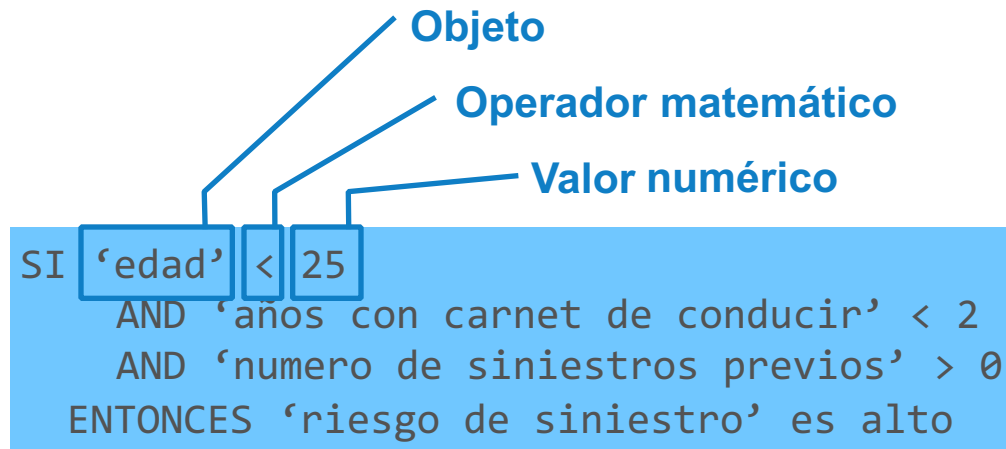
- ▶ Estructuras más complejas:
 - Múltiples condiciones unidas por operadores lógicos
 - AND (conjunción)
 - OR (disyunción)
 - Múltiples conclusiones

```
SI <antecedente 1>  
    AND <antecedente 2>  
    AND <antecedente 3>  
    ENTONCES <consecuente 1>  
              <consecuente 2>
```

Representación del conocimiento mediante reglas

► Estructura del antecedente

```
SI <objeto> <operador> <valor>  
  ENTONCES <consecuente>
```



► Tipos de operadores

- Matemáticos → Valor numérico del objeto
- Lingüísticos → Valor lingüístico del objeto

Representación del conocimiento mediante reglas

► Estructura del consecuente

– Opción 1

```
SI <antecedente>  
  ENTONCES <objeto> <operador> <valor>
```

– Opción 2

```
SI <antecedente>  
  ENTONCES <objeto> = <expresión aritmética>
```

```
SI 'edad' < 25  
  AND 'años con carnet de conducir' < 2  
  AND 'numero de siniestros previos' > 0  
  ENTONCES 'riesgo de siniestro' es alto
```

Objeto

Operador lingüístico

Valor lingüístico

```
SI 'edad' < 25  
  AND 'años con carnet de conducir' < 2  
  AND 'numero de siniestros previos' > 0  
  ENTONCES 'riesgo de siniestro' = 'edad' * 1,5
```

Objeto

Expresión aritmética

Reglas de clasificación y Reglas de asociación

SI <antecedente>
ENTONCES <consecuente>

REGLAS DE ASOCIACIÓN

- Predicen valores de atributos, combinaciones de valores de atributos, o la propia clase
- Descubrir combinaciones de pares atributo-valor que ocurren con cierta frecuencia

Restricciones de valores de los atributos

SI *temperatura=media*
ENTONCES *humedad = alta*

Cualquier combinación de valores de atributos

REGLAS DE CLASIFICACIÓN

- Predicen la clase
- Clasificar futuras instancias

Restricciones de valores de los atributos

SI *humedad=alta*
AND *ambiente =soleado*
ENTONCES *jugar = No*

Valor de la clase

Reglas de clasificación y Reglas de asociación

SI <antecedente>
ENTONCES <consecuente>

REGLAS DE ASOCIACIÓN

¿Qué productos compran conjuntamente los clientes del supermercado?



Fuente:
https://commons.wikimedia.org/wiki/File:Shopping_cart_with_food_clip_art_2.svg

REGLAS DE CLASIFICACIÓN

¿Qué enfermedad se le diagnosticaría a un paciente que tiene ciertos síntomas?



Fuente:
<https://pixabay.com/en/prognosis-icon-patient-chart-2803190/>

Reglas de clasificación y Reglas de asociación

Específicamente dos medidas populares
que se suelen utilizar con el fin de evaluar una regla son:

Confianza

$$\text{Confianza } (A \rightarrow B) = P(B|A)$$

- Probabilidad de que el consecuente de la regla se cumpla si se da el antecedente
- Proporción de ejemplos que satisfacen la regla (cumplen el antecedente y consecuente) respecto a aquellos ejemplos que sólo satisfacen el antecedente
- Mide el **interés** de la regla

Soporte

$$\text{Soporte } (A \rightarrow B) = P(A \cap B)$$

- Probabilidad de que se cumpla el antecedente y el consecuente de la regla
- Proporción de ejemplos que satisfacen la regla (cumplen el antecedente y el consecuente) respecto al número total de ejemplos
- Mide la **frecuencia** de la regla

Reglas de clasificación y Reglas de asociación

Específicamente dos medidas populares
que se suelen utilizar con el fin de evaluar una regla son:

Confianza

Soporte

SI detergente
ENTONCES suavizante



*Probabilidad de que la compra
de un detergente conduzca a la
compra de un suavizante*

*Frecuencia con la que se han
comprado a la vez detergente y
suavizante*

Ejemplo:

- Se están analizando 1 000 cestas de la compra
- 100 cestas contienen detergente y suavizante
- 20 cestas contienen detergente pero no suavizante

$$\text{Confianza} = 100/120 = 0.833$$

$$\text{Soporte} = 100/1000 = 0.1$$

Reglas de clasificación y Reglas de asociación

Específicamente dos medidas populares que se suelen utilizar con el fin de evaluar una regla son:

Confianza

Soporte

SI detergente
ENTONCES suavizante



Probabilidad de que la compra de un detergente conduzca a la compra de un suavizante

Frecuencia con la que se han comprado a la vez detergente y suavizante

Ejemplo:

- Se están analizando 1 000 cestas de la compra
- 1 cesta contiene detergente y suavizante
- No hay ninguna otra cesta que contenga detergente

Regla irrelevante:
tiene alta confianza
pero cubre casos
poco frecuentes

$$\text{Confianza} = 1/1 = 1$$

$$\text{Soporte} = 1/1000 = 0,001$$

Reglas de clasificación y Reglas de asociación

Específicamente dos medidas populares
que se suelen utilizar con el fin de evaluar una regla son:

Confianza

Soporte

⇩ Sin embargo otras métricas son necesarias

Lift

$$lift(A, B) = \frac{P(B|A)}{P(B)} = \frac{confianza(A \rightarrow B)}{P(B)}$$

- Relación entre la probabilidad de que el consecuente de la regla se cumpla si se da el antecedente y la probabilidad de que se cumpla el consecuente de la regla
- Mide la **correlación** entre la ocurrencia de un hecho A y un hecho B
 - $lift = 1$ A es independiente de B
 - $lift > 1$ correlación entre A y B \rightarrow A probablemente implica B [regla útil]
 - $lift < 1$ correlación negativa entre A y B \rightarrow A probablemente implica $\neg B$

Reglas de clasificación y Reglas de asociación

Específicamente dos medidas populares
que se suelen utilizar con el fin de evaluar una regla son:

Confianza

Soporte

⇩ Sin embargo otras métricas son necesarias

Lift



Ejemplo (Han et al, 2012):

- Se analizan 10 000 transacciones
- 6 000 contienen juegos de ordenador
- 7 500 incluyen vídeos → Probabilidad de comprar vídeos es de 0.75
- 4 000 incluyen tanto vídeos como juegos de ordenador

Han, J., Kamber, M., & Pei, J. (2012).
Data mining: concepts and technique.
San Francisco: Morgan Kaufmann.

‘Juegos de ordenador’ → ‘videos’ (soporte 0.4, confianza 0.66)

$$\text{Confianza} = 4000/6000 = 0,66$$

$$\text{Soporte} = 4000/10000 = 0.4$$

$$\text{Lift} = 0,66/0.75 = 0.88 < 1 \rightarrow \text{Juegos y videos compiten entre sí}$$

Reglas de clasificación y Reglas de asociación

- ▶ Ejemplo: Problema "Jugar al aire libre", J.R. Quinlan (1986)

REGLAS DE ASOCIACIÓN

Regla 1:

*Si Ambiente es nublado
ENTONCES Jugar = sí*

Regla 2:

*Si Temperatura es baja
ENTONCES Humedad es normal*

Regla 3:

*Si Temperatura es media
ENTONCES Humedad es alta*

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

REGLA DE CLASIFICACIÓN

Reglas de clasificación y Reglas de asociación

- Ejemplo: Problema "Jugar al aire libre", J.R. Quinlan (1986)

Regla 3:

*Si Temperatura es media
ENTONCES Humedad es alta*

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

$$\text{Confianza } (\text{Temperatura} = \text{media} \rightarrow \text{Humedad} = \text{alta}) = \frac{4}{6} = 0,67$$

Reglas de clasificación y Reglas de asociación

- ▶ Ejemplo: Problema "Jugar al aire libre", J.R. Quinlan (1986)

Regla 3:

*Si Temperatura es media
ENTONCES Humedad es alta*

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

$$\text{Soporte (temperatura = media} \rightarrow \text{humedad = alta)} = \frac{4}{14} = 0,29$$

Reglas de clasificación y Reglas de asociación

- Ejemplo: Problema "Jugar al aire libre", J.R. Quinlan (1986)

Regla 3:

*Si Temperatura es media
ENTONCES Humedad es alta*

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

$$\text{Lift (Temperatura = media} \rightarrow \text{Humedad = alta)} = \frac{4/6}{7/14} = 1,33$$

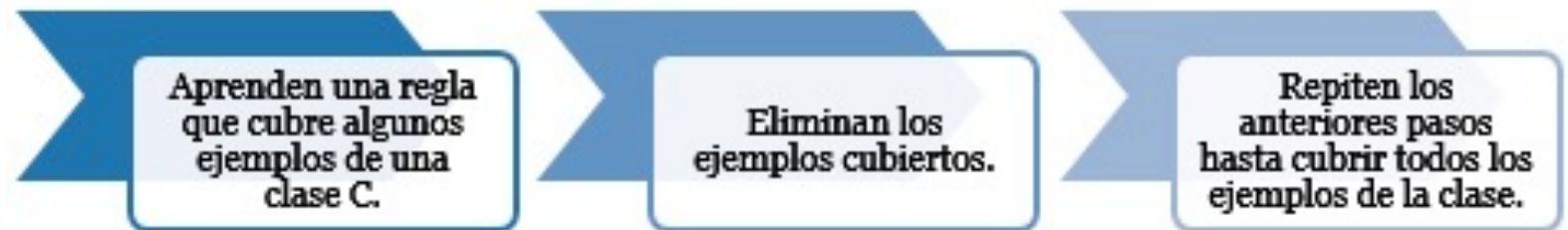


Algoritmos de aprendizaje de reglas de clasificación

▶ Algoritmos de recubrimiento secuencial

- Aprendizaje directo de conjuntos de reglas de clasificación

Estos algoritmos en cada iteración:



- Uso iterativo de un procedimiento que seleccione una única regla de buena precisión pero sin necesidad de que cubra todos los ejemplos positivos



Algoritmos de aprendizaje de reglas de clasificación

► Algoritmo básico de recubrimiento secuencial

```
PROCEDIMIENTO Recubrimiento_secuencial (Clases, atributos, ejemplos)
COMIENZO
  Reglas  $\leftarrow \{\}$ 
  Para cada clase C de Clases
    COMIENZO
      E  $\leftarrow$  Ejemplos
      Mientras (E contenga ejemplos de la clase C)
        COMIENZO
          Regla  $\leftarrow$  AprenderUnaRegla (C, E, Atributos)
          Reglas  $\leftarrow$  Reglas + {Regla}
          E  $\leftarrow$  E - {Ejemplos de E clasificados correctamente por Regla}
        FIN
      FIN
    Devolver Reglas
  FIN
```

- Se podría incluir una condición adicional que evalúe la calidad de la regla aprendida para considerar o descartar esta regla



Algoritmos de aprendizaje de reglas de clasificación

► Algoritmo básico de aprendizaje de una regla

```
PROCEDIMIENTO AprenderUnaRegla (Clase, Ejemplos, Atributos)
COMIENZO
  Regla ← regla con antecedente A vacío y con consecuente Clase
  MIENTRAS (regla cubre algún ejemplo negativo AND Atributos ≠ ∅)
  COMIENZO
    Restricciones ← {}
    Para cada atributo A no utilizado en la regla
    COMIENZO
      Para cada valor v de A
      COMIENZO
        Restricciones ← Restricciones + {A=v}
      FIN
    FIN
    Restriccion ← mejorRestriccion (Restricciones, regla)
    Regla ← añadir restricción al antecedente
    Atributos ← atributos - {atributo de Restriccion}
  FIN
  Devolver Regla
FIN
```

— Búsqueda codiciosa (sin retroceso) de lo general a lo específico



Algoritmos de aprendizaje de reglas de clasificación

- ▶ **Algoritmo PRISM** (Cendrowska, 1987)
 - Algoritmo de recubrimiento secuencial
 - Aprendizaje de las reglas basado en la medida de **confianza** (utilizada en el procedimiento *mejorRestriccion*)
 - Lógica del algoritmo:
 - Se escoge la regla más general (sin restricciones en el antecedente)
 - Se selecciona la mejor restricción en base a la confianza
 - En caso de empate se escoge la restricción de mayor cobertura (o mayor soporte)



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM



PROCEDIMIENTO Recubrimiento_secuencial (Clases, atributos, ejemplos)

COMIENZO

Reglas $\leftarrow \{\}$

Para cada clase C de Clases **C= No**

COMIENZO

E \leftarrow Ejemplos **E = {E1, E2, ... E14}**

Mientras (E contenga ejemplos de la clase C) **VERDADERO**

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No

COMIENZO

Regla \leftarrow AprenderUnaRegla (C, E, Atributos)



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

No {E1, E2, ... E14} {Ambiente, Temperatura, Humedad, Viento}
 ↓ ↓ ↓
 PROCEDIMIENTO AprenderUnaRegla (Clase, Ejemplos, Atributos)
 COMIENZO
 Regla ← regla con antecedente A vacío y con consecuente Clase
 SI --- ENTONCES jugar = No
 MIENTRAS (regla cubre algún ejemplo negativo AND Atributos ≠ ∅) VERDADERO
 COMIENZO
 Restricciones ← {}
 Para cada atributo A no utilizado en la regla A = Ambiente
 COMIENZO
 Para cada valor v de A v = soleado
 COMIENZO
 Restricciones ← Restricciones + {A=v}
 Restricciones = {Ambiente = soleado}
 FIN
 FIN
 ...
 Restricciones = {Ambiente=soleado, Ambiente=nublado, Ambiente=lluvioso, Temperatura=alta, Temperatura=media, Temperatura=baja, Humedad=alta, Humedad=normal, Viento=verdadero, Viento = falso}
 Restriccion ← mejorRestriccion (Restricciones, regla)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	lluvioso	Media	Alta	Falso	Si
E5	lluvioso	Baja	Normal	Falso	Si
E6	lluvioso	Baja	Normal	Verdadero	No
E7	nublado	Baja	Normal	Verdadero	Si
E8	soleado	Media	Alta	Falso	No
E9	soleado	Baja	Normal	Falso	Si
E10	lluvioso	Media	Normal	Falso	Si
E11	soleado	Media	Normal	Verdadero	Si
E12	nublado	Media	Alta	Verdadero	Si
E13	nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Restricciones = {Ambiente=soleado, Ambiente=nublado, Ambiente=lluvioso, Temperatura=alta, Temperatura=media, Temperatura=baja, Humedad=alta, Humedad=normal, Viento=verdadero, Viento = falso}

SI --- ENTONCES jugar = No

Restriccion ← mejorRestriccion (Restricciones, regla)

SI Ambiente=soleado ENTONCES jugar = No

Confianza = $3/5 = 0,6$

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Restricciones = {Ambiente=soleado, Ambiente=nublado,
Ambiente=lluvioso, Temperatura=alta, Temperatura=media,
Temperatura=baja, Humedad=alta, Humedad=normal,
Viento=verdadero, Viento = falso}

SI --- ENTONCES jugar = No

Restriccion ← mejorRestriccion (Restricciones, regla)

SI Ambiente=soleado ENTONCES jugar = No

SI Ambiente=nublado ENTONCES jugar = No

SI Ambiente=lluvioso ENTONCES jugar = No

SI Temperatura=alta ENTONCES jugar = No

SI Temperatura=media ENTONCES jugar = No

SI Temperatura=baja ENTONCES jugar = No

SI Humedad=alta ENTONCES jugar = No

SI Humedad=normal ENTONCES jugar = No

SI Viento=verdadero ENTONCES jugar = No

SI Viento = falso ENTONCES jugar = No

Confianza = 0,6

Confianza = 0

Confianza = 0,4

Confianza = 0,5

Confianza = 0,33

Confianza = 0,25

Confianza = 0,57

Confianza = 0,14

Confianza = 0,5

Confianza = 0,25

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No

Restricción = Ambiente=soleado



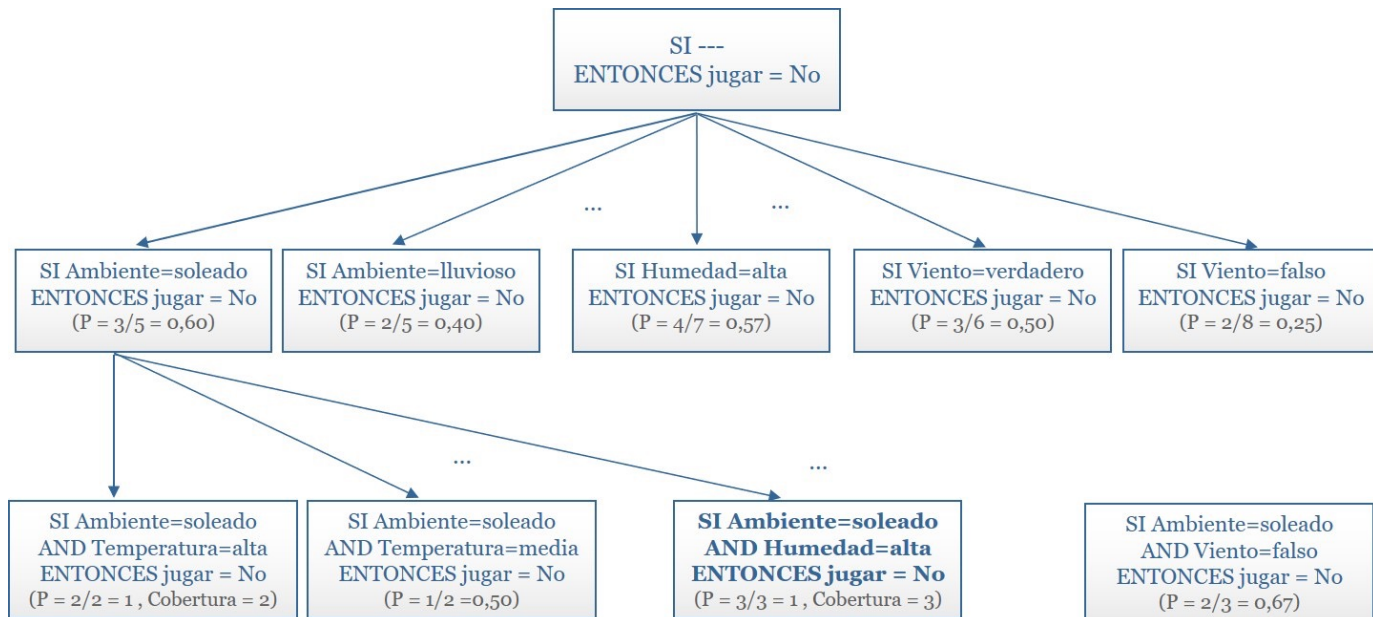
Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Restricciones = {Ambiente=soleado, Ambiente=nublado, Ambiente=lluvioso, Temperatura=alta, Temperatura=media, Temperatura=baja, Humedad=alta, Humedad=normal, Viento=verdadero, Viento = falso}

SI --- ENTONCES jugar = No

Restriccion ← mejorRestriccion (Restricciones, regla)





Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

PROCEDIMIENTO AprenderUnaRegla (Clase, Ejemplos, Atributos)

COMIENZO

Regla \leftarrow regla con antecedente A vacío y con consecuente Clase

MIENTRAS (regla cubre algún ejemplo negativo AND Atributos $\neq \emptyset$)

COMIENZO

Restricciones $\leftarrow \{\}$

Para cada atributo A no utilizado en la regla

COMIENZO

Para cada valor v de A

COMIENZO

Restricciones \leftarrow Restricciones + {A=v}

FIN

FIN

Restriccion \leftarrow mejorRestriccion (Restricciones, regla)

Ambiente=soleado

Regla \leftarrow añadir restricción al antecedente

SI Ambiente=soleado ENTONCES jugar = No

Atributos \leftarrow atributos - {atributo de Restriccion}

Atributos = {Temperatura, Humedad, Viento}

FIN



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

PROCEDIMIENTO AprenderUnaRegla (Clase, Ejemplos, Atributos)

COMIENZO

Regla \leftarrow regla con antecedente A vacío y con consecuente Clase

MIENTRAS (regla cubre algún ejemplo negativo AND Atributos $\neq \emptyset$)

Regla: SI Ambiente=soleado ENTONCES jugar = No

Atributos = {Temperatura, Humedad, Viento}

=> VERDADERO

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No

COMIENZO

Restricciones $\leftarrow \{\}$

Para cada atributo A no utilizado en la regla

COMIENZO

Para cada valor v de A

COMIENZO

Restricciones \leftarrow Restricciones + {A=v}

FIN

FIN

Restricciones = {Temperatura=alta, Temperatura=media, Temperatura=baja, Humedad=alta, Humedad=normal, Viento=verdadero, Viento = falso}

Restriccion \leftarrow mejorRestriccion (Restricciones, regla)



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Restricciones = {Temperatura=alta, Temperatura=media,
Temperatura=baja, Humedad=alta, Humedad=normal,
Viento=verdadero, Viento = falso}

SI Ambiente=soleado
ENTONCES jugar = No

Restriccion ← mejorRestriccion (Restricciones, regla)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No

SI Ambiente=soleado AND Temperatura=alta ENTONCES jugar = No

SI Ambiente=soleado AND Temperatura=media ENTONCES jugar = No

SI Ambiente=soleado AND Temperatura=baja ENTONCES jugar = No

SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No

SI Ambiente=soleado AND Humedad=normal ENTONCES jugar = No

SI Ambiente=soleado AND Viento=verdadero ENTONCES jugar = No

SI Ambiente=soleado AND Viento = falso ENTONCES jugar = No

Confianza = 1

Confianza = 0,5

Confianza = 0

Confianza = 1

Confianza = 0

Confianza = 0,5

Confianza = 0,67

SI Ambiente=soleado AND Temperatura=alta ENTONCES jugar = No

SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No

Cobertura = 2

Cobertura = 3

Restricción = Humedad=alta



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

PROCEDIMIENTO AprenderUnaRegla (Clase, Ejemplos, Atributos)

COMIENZO

Regla \leftarrow regla con antecedente A vacío y con consecuente Clase

MIENTRAS (regla cubre algún ejemplo negativo AND Atributos $\neq \emptyset$)

COMIENZO

Restricciones $\leftarrow \{\}$

Para cada atributo A no utilizado en la regla

COMIENZO

Para cada valor v de A

COMIENZO

Restricciones \leftarrow Restricciones + {A=v}

FIN

FIN

Restriccion \leftarrow mejorRestriccion (Restricciones, regla)

Humedad=alta

Regla \leftarrow añadir restricción al antecedente

SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No

Atributos \leftarrow atributos - {atributo de Restriccion}

Atributos = {Temperatura, Viento}

FIN



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

PROCEDIMIENTO AprenderUnaRegla (Clase, Ejemplos, Atributos)

COMIENZO

Regla \leftarrow regla con antecedente A vacío y con consecuente Clase

MIENTRAS (regla cubre algún ejemplo negativo AND Atributos $\neq \emptyset$) \Rightarrow FALSO

Regla: SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No

Atributos = {Temperatura, Viento}

COMIENZO

Restricciones $\leftarrow \{\}$

Para cada atributo A no utilizado en la regla

COMIENZO

Para cada valor v de A

COMIENZO

Restricciones \leftarrow Restricciones + {A=v}

FIN

FIN

Restriccion \leftarrow mejorRestriccion (Restricciones, regla)

Regla \leftarrow añadir restricción al antecedente

Atributos \leftarrow atributos - {atributo de Restriccion}

FIN

Devolver Regla

FIN

SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	Lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

PROCEDIMIENTO Recubrimiento_secuencial (Clases, atributos, ejemplos)

COMIENZO

Reglas $\leftarrow \{\}$

Para cada clase C de Clases C= No

COMIENZO

E \leftarrow Ejemplos E = {E1, E2, ... E14}

Mientras (E contenga ejemplos de la clase C)

COMIENZO

Regla \leftarrow AprenderUnaRegla (C, E, Atributos)

SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No

Reglas \leftarrow Reglas + {Regla}

Reglas = {SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No}

E \leftarrow E - {Ejemplos de E clasificados correctamente por Regla}

E = {E3, ... E7, E9, ... E14}

FIN

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

PROCEDIMIENTO Recubrimiento_secuencial (Clases, atributos, ejemplos)

COMIENZO

Reglas $\leftarrow \{\}$

Para cada clase C de Clases $C = \text{No}$

COMIENZO

E \leftarrow Ejemplos

Mientras (E contenga ejemplos de la clase C)

$E = \{E3, \dots E7, E9, \dots E14\} \Rightarrow \text{Verdadero}$

COMIENZO

Regla \leftarrow AprenderUnaRegla (C, E, Atributos)

Regla \leftarrow AprenderUnaRegla (No, $\{E3, \dots E7, E9, \dots E14\}$, {Ambiente, Temperatura, Humedad, Viento})

...

Reglas \leftarrow Reglas + {Regla}

E \leftarrow E - {Ejemplos de E clasificados correctamente por Regla}

FIN

FIN

Devolver Reglas

Reglas = {SI Ambiente=soleado AND Humedad=alta ENTONCES jugar = No,
SI ...}

FIN

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Aplicación del algoritmo PRISM

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No



Algoritmos de aprendizaje de reglas de clasificación

► Ejemplo PRISM en Weka

=== Run information ===

Scheme: weka.classifiers.rules.Prism

Relation: weather.symbolic

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Prism rules

If outlook = overcast then yes

If humidity = normal

and windy = FALSE then yes

If temperature = mild

and humidity = normal then yes

If outlook = rainy

and windy = FALSE then yes

If outlook = sunny

and humidity = high then no

If outlook = rainy

and windy = TRUE then no

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9	64.2857 %
Incorrectly Classified Instances	3	21.4286 %
Kappa statistic	0.4375	
Mean absolute error	0.25	
Root mean squared error	0.5	
Relative absolute error	59.2264 %	
Root relative squared error	105.9121 %	
UnClassified Instances	2	14.2857 %
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,600	0,700	1,000	0,824	0,529	0,589	0,687	yes
	0,400	0,000	1,000	0,400	0,571	0,529	0,700	0,614	no
Weighted Avg.	0,750	0,350	0,825	0,750	0,718	0,529	0,635	0,657	

=== Confusion Matrix ===

a b <-- classified as

7 0 | a = yes

3 2 | b = no

Algoritmos de aprendizaje de reglas de asociación

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No



REGLAS DE ASOCIACIÓN

*Si Ambiente es nublado
ENTONCES Jugar = sí*

*Si Temperatura es baja
ENTONCES Humedad es normal*

*Si Temperatura es media
ENTONCES Humedad es alta*

...

► Fases del aprendizaje de reglas de asociación

1. Encontrar aquellas reglas cuya frecuencia sea superior a un valor de soporte establecido.
2. De las reglas extraídas en el paso 1, seleccionar aquellas cuya confianza es superior a un valor determinado.



Algoritmos de aprendizaje de reglas de asociación

► Algoritmo Apriori

- Pretende generar **ítem-sets** que cumplan una **cobertura** mínima
- Definiciones:
 - **Ítem**: Par atributo-valor
 - **Ítem-set**: conjunto de pares atributo-valor
 - **k-ítem-set**: conjunto de k pares atributo-valor
 - **Cobertura**: número de instancias que cumplen los valores en los ítem-sets

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1



Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

- ▶ Fase 0: Determinar la cobertura mínima
 - Ej. Ítem-set de Cobertura = 3 ↔ Regla con Soporte = $3/14 = 0.21$
- ▶ Fase 1: Generar k -ítem-sets escogiendo los que cumplen la cobertura mínima
 - Iterar en k (número de pares atributo-valor en el ítem-set)

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Primera iteración: 1-ítem-sets
 - Generar ítem-sets de 1 par atributo-valor

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No



Ambiente = soleado
 Ambiente = lluvioso
 Ambiente = nublado
 Temperatura = alta
 Temperatura = media
 ...
 Jugar = Sí
 Jugar = No

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Primera iteración: 1-ítem-sets
 - Calcular la cobertura de los ítem-sets

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

Ambiente = soleado

⇒ Cobertura = 5

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Primera iteración: 1-ítem-sets
 - Calcular la cobertura de los ítem-sets

Id	Ambiente	Temperatura	Humedad	Viento	Jugar		Item-sets de 1 elemento	Cobertura
E1	soleado	Alta	Alta	Falso	No	➔	Ambiente=soleado	5
E2	soleado	Alta	Alta	Verdadero	No		Ambiente=nublado	4
E3	nublado	Alta	Alta	Falso	Sí		Ambiente=lluvioso	5
E4	Lluvioso	Media	Alta	Falso	Sí		Temperatura=alta	4
E5	Lluvioso	Baja	Normal	Falso	Sí		Temperatura=media	6
E6	Lluvioso	Baja	Normal	Verdadero	No		Temperatura=baja	4
E7	Nublado	Baja	Normal	Verdadero	Sí		Humedad=alta	7
E8	Soleado	Media	Alta	Falso	No		Humedad=normal	7
E9	Soleado	Baja	Normal	Falso	Sí		Viento=falso	8
E10	Lluvioso	Media	Normal	Falso	Sí		Viento=verdadero	6
E11	Soleado	Media	Normal	Verdadero	Sí		Jugar=sí	9
E12	Nublado	Media	Alta	Verdadero	Sí		Jugar=no	5
E13	Nublado	Alta	Normal	Falso	Sí			
E14	lluvioso	Media	alta	Verdadero	No			

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Primera iteración: 1-ítem-sets
 - Escoger los ítem-sets que cumplen la cobertura mínima (3)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

Item-sets de 1 elemento	Cobertura
Ambiente=soleado	5
Ambiente=nublado	4
Ambiente=lluvioso	5
Temperatura=alta	4
Temperatura=media	6
Temperatura=baja	4
Humedad=alta	7
Humedad=normal	7
Viento=falso	8
Viento=verdadero	6
Jugar=sí	9
Jugar=no	5

Todos tienen la mínima cobertura

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Segunda iteración: 2-ítem-sets
 - Generar *ítem-sets* de 2 pares atributo-valor combinando los *ítem-sets* encontrados en la primera iteración

Ítem-sets de 1 elemento

Ambiente=soleado
Ambiente=nublado
Ambiente=lluvioso
Temperatura=alta
Temperatura=media
Temperatura=baja
Humedad=alta
Humedad=normal
Viento=falso
Viento=verdadero
Jugar=si
Jugar=no



Ambiente = soleado, Temperatura = alta
Ambiente = soleado, Temperatura = media
Ambiente = soleado, Temperatura = baja
Ambiente = soleado, Humedad = alta
Ambiente = soleado, Humedad = normal
Ambiente = soleado, Viento = falso
Ambiente = soleado, Viento = verdadero
Ambiente = soleado, Jugar = Sí
Ambiente = soleado, Jugar = No
...
Viento = verdadero, Jugar = No

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Segunda iteración: 2-ítem-sets
 - Calcular la cobertura de los ítem-sets y escoger los ítem-sets que cumplen la cobertura mínima (3)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

~~Ambiente = soleado, Temperatura = alta~~

⇒ Cobertura = 2

¡No cumple la cobertura mínima!

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

– Segunda iteración: 2-ítem-sets

- Calcular la cobertura de los ítem-sets y escoger los ítem-sets que cumplen la cobertura mínima (3)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

Ambiente = soleado, Humedad = alta

⇒ Cobertura = 3

Cumple la cobertura mínima
→ se selecciona el ítem-set

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Segunda iteración: 2-ítem-sets
 - Calcular la cobertura de los ítem-sets y escoger los ítem-sets que cumplen la cobertura mínima (3)

<i>Item-sets de 2 elementos</i>	<i>Cobertura</i>	<i>Item-sets de 2 elementos</i>	<i>Cobertura</i>
Ambiente=lluvioso, temperatura=media	3	Temperatura=media, jugar=si	4
Ambiente=lluvioso, humedad=normal	3	Humedad=alta, viento=falso	4
Ambiente=soleado, humedad=alta	3	Humedad=alta, viento=verdadero	3
Ambiente=lluvioso, viento=falso	3	Humedad=normal, viento=falso	4
Ambiente=soleado, viento=falso	3	Humedad=normal, viento=verdadero	3
Ambiente=lluvioso, jugar=si	3	Humedad=normal, jugar=si	6
Ambiente=nublado, jugar=si	4	Humedad=alta, jugar=si	3
Ambiente=soleado, jugar=no	3	Humedad=alta, jugar=no	4
Temperatura=alta, humedad=alta	3	Viento=falso, jugar=si	6
Temperatura=baja, humedad=normal	4	Viento=verdadero, jugar=si	3
Temperatura=media, humedad=alta	4	Viento=verdadero, jugar=no	3
Temperatura=alta, viento=falso	3		
Temperatura=media, viento=falso	3		
Temperatura=media, viento=verdadero	3		
Temperatura=baja, jugar=si	3		

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

– Tercera iteración: 3-ítem-sets

- Generar *ítem-sets* de 3 pares atributo-valor combinando los *ítem-sets* de dos elementos

<i>Item-sets</i> de 2 elementos	Cobertura	<i>Item-sets</i> de 2 elementos	Cobertura
Ambiente=lluvioso, temperatura=media	3	Temperatura=media, jugar=si	4
Ambiente=lluvioso, humedad=normal	3	Humedad=alta, viento=falso	4
Ambiente=soleado, humedad=alta	3	Humedad=alta, viento=verdadero	3
Ambiente=lluvioso, viento=falso	3	Humedad=normal, viento=falso	4
Ambiente=soleado, viento=falso	3	Humedad=normal, viento=verdadero	3
Ambiente=lluvioso, jugar=si	3	Humedad=normal, jugar=si	6
Ambiente=nublado, jugar=si	4	Humedad=alta, jugar=si	3
Ambiente=soleado, jugar=no	3	Humedad=alta, jugar=no	4
Temperatura=alta, humedad=alta	3	Viento=falso, jugar=si	6
Temperatura=baja, humedad=normal	4	Viento=verdadero, jugar=si	3
Temperatura=media, humedad=alta	4	Viento=verdadero, jugar=no	3
Temperatura=alta, viento=falso	3		
Temperatura=media, viento=falso	3		
Temperatura=media, viento=verdadero	3		
Temperatura=baja, jugar=si	3		



Ambiente = lluvioso,
Temperatura = media,
Humedad = normal

Ambiente = soleado,
Humedad = alta,
Jugar = no

...

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Tercera iteración: 3-ítem-sets
 - Calcular la cobertura de los ítem-sets y escoger los ítem-sets que cumplen la cobertura mínima (3)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

~~Ambiente = lluvioso,
Temperatura = media,
Humedad = normal~~

⇒ Cobertura = 1

¡No cumple la cobertura mínima!

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Tercera iteración: 3-ítem-sets
 - Calcular la cobertura de los ítem-sets y escoger los ítem-sets que cumplen la cobertura mínima (3)

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

Ambiente = soleado,
Humedad = alta,
Jugar = no

⇒ Cobertura = 3

Cumple la cobertura mínima
→ se selecciona el ítem-set



Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Tercera iteración: 3-ítem-sets
 - Calcular la cobertura de los ítem-sets y escoger los ítem-sets que cumplen la cobertura mínima (3)

Item-sets de 3 elementos	Cobertura
Ambiente=soleado, humedad=alta, jugar=no	3
Ambiente=lluvioso, viento=falso, jugar=si	3
Temperatura=baja, humedad=normal, jugar=si	3
Humedad=normal, viento=falso, jugar=si	4



Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Cuarta iteración: 4-ítem-sets
 - Generar *ítem-sets* de 4 pares atributo-valor combinando los *ítem-sets* de tres elementos

Item-sets de 3 elementos	Cobertura
Ambiente=soleado, humedad=alta, jugar=no	3
Ambiente=lluvioso, viento=falso, jugar=si	3
Temperatura=baja, humedad=normal, jugar=si	3
Humedad=normal, viento=falso, jugar=si	4



No hay combinaciones de 4 elementos → **fin FASE 1**

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Fase 2: Generar reglas de asociación a partir de los *ítem-sets*, descartando las reglas que no superan un mínimo valor de confianza

Ej. Confianza = 0.9

Item-sets de 1 elemento

Ambiente=soleado
Ambiente=nublado
Ambiente=lluvioso
Temperatura=alta
Temperatura=media
Temperatura=baja
Humedad=alta
Humedad=normal
Viento=falso
Viento=verdadero
Jugar=si
Jugar=no

Item-sets de 2 elementos

Ambiente=lluvioso, temperatura=media
Ambiente=lluvioso, humedad=normal
Ambiente=soleado, humedad=alta
Ambiente=lluvioso, viento=falso
Ambiente=soleado, viento=falso
Ambiente=lluvioso, jugar=si
Ambiente=nublado, jugar=si
Ambiente=soleado, jugar=no
Temperatura=alta, humedad=alta
Temperatura=baja, humedad=normal
Temperatura=media, humedad=alta
Temperatura=alta, viento=falso
Temperatura=media, viento=falso
Temperatura=media, viento=verdadero
Temperatura=baja, jugar=si

Item-sets de 2 elementos

Temperatura=media, jugar=si
Humedad=alta, viento=falso
Humedad=alta, viento=verdadero
Humedad=normal, viento=falso
Humedad=normal, viento=verdadero
Humedad=normal, jugar=si
Humedad=alta, jugar=si
Humedad=alta, jugar=no
Viento=falso, jugar=si
Viento=verdadero, jugar=si
Viento=verdadero, jugar=no

Item-sets de 3 elementos

Ambiente=soleado, humedad=alta, jugar=no
Ambiente=lluvioso, viento=falso, jugar=si
Temperatura=baja, humedad=normal, jugar=si
Humedad=normal, viento=falso, jugar=si

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Generar reglas de asociación a partir de los *ítem-sets*

Item-sets de 3 elementos	Cobertura
Ambiente=soleado, humedad=alta, jugar=no	3



SI ambiente = soleado
 ENTONCES humedad=alta AND jugar=no
 SI ambiente = soleado AND humedad=alta
 ENTONCES jugar=no
 SI ambiente = soleado AND jugar=no
 ENTONCES humedad=alta
 SI humedad=alta
 ENTONCES jugar=no AND ambiente=soleado
 SI humedad=alta AND jugar= no
 ENTONCES ambiente=soleado
 SI jugar=no
 ENTONCES humedad=alta AND ambiente=soleado

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Calcular la confianza de las reglas

SI ambiente = soleado
ENTONCES humedad=alta AND jugar=no

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No

$$\text{Confianza} = \frac{3}{5} = 0,60$$

¡No cumple la confianza mínima!

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Calcular la confianza de las reglas

SI ambiente = soleado AND humedad=alta
ENTONCES jugar=no

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Si
E4	Lluvioso	Media	Alta	Falso	Si
E5	Lluvioso	Baja	Normal	Falso	Si
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Si
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Si
E10	Lluvioso	Media	Normal	Falso	Si
E11	Soleado	Media	Normal	Verdadero	Si
E12	Nublado	Media	Alta	Verdadero	Si
E13	Nublado	Alta	Normal	Falso	Si
E14	lluvioso	Media	alta	Verdadero	No

$$\text{Confianza} = \frac{3}{3} = 1$$

Cumple la confianza mínima

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Calcular la confianza de las reglas

Item-sets de 3 elementos	Cobertura
Ambiente=soleado, humedad=alta, jugar=no	3



SI ambiente = soleado ENTONCES humedad=alta AND jugar=no	Conf = $3/5 = 0.6$
SI ambiente = soleado AND humedad=alta ENTONCES jugar=no	Conf = $3/3 = 1$
SI ambiente = soleado AND jugar=no ENTONCES humedad=alta	Conf = $3/3 = 1$
SI humedad=alta ENTONCES jugar=no AND ambiente=soleado	Conf = $3/7 = 0.43$
SI humedad=alta AND jugar= no ENTONCES ambiente=soleado	Conf = $3/4 = 0.75$
SI jugar=no ENTONCES humedad=alta AND ambiente=soleado	Conf = $3/5 = 0.6$

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Descartar las reglas que no superan un mínimo valor de confianza

Confianza mínima = 0.9

SI ambiente = soleado	Conf: 0.6
ENTONCES humedad=alta AND jugar=no	
SI ambiente = soleado AND humedad=alta	Conf: 1
ENTONCES jugar=no	
SI ambiente = soleado AND jugar=no	Conf: 1
ENTONCES humedad=alta	
SI humedad=alta	Conf: 0.43
ENTONCES jugar=no AND ambiente=soleado	
SI humedad=alta AND jugar= no	Conf: 0.75
ENTONCES ambiente=soleado	
SI jugar=no	Conf: 0.6
ENTONCES humedad=alta AND ambiente=soleado	

Algoritmos de aprendizaje de reglas de asociación

El algoritmo apriori utiliza dos fases:

FASE 1: Generación de ítem-sets

FASE 2: Generación de reglas a partir de los ítems-sets generados en la fase 1

- Repetir el proceso para cada uno de los ítem-sets
- Conjunto final de reglas

SI ambiente = soleado AND humedad = alta	
ENTONCES jugar = no	Conf: 1
SI ambiente = soleado AND jugar = no	
ENTONCES humedad = alta	Conf: 1
SI ambiente = nublado	
ENTONCES jugar = sí	Conf: 1
SI temperatura = baja	
ENTONCES humedad= normal	Conf: 1
SI humedad = normal AND viento= falso	
ENTONCES jugar = si	Conf: 1
SI ambiente = lluvioso AND viento = falso	
ENTONCES jugar = si	Conf: 1
SI ambiente = lluvioso AND jugar = si	
ENTONCES viento = falso	Conf: 1
SI temperatura = baja AND jugar = sí	
ENTONCES humedad = normal	Conf: 1



Algoritmos de aprendizaje de reglas de asociación

► Ejemplo Apriori en Weka

=== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.21 -S -1.0 -c -1

Relation: weather.symbolic

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

=== Associator model (full training set) ===

Apriori

=====

Minimum support: 0.21 (3 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 26

Size of set of large itemsets L(3): 4

Best rules found:

1. outlook=overcast 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
2. temperature=cool 4 ==> humidity=normal 4 <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
3. humidity=normal windy=FALSE 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
4. outlook=sunny play=no 3 ==> humidity=high 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
5. outlook=sunny humidity=high 3 ==> play=no 3 <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
7. outlook=rainy windy=FALSE 3 ==> play=yes 3 <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
8. temperature=cool play=yes 3 ==> humidity=normal 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)

Ejemplo práctico de aplicación (apriori)



- ▶ Aplicación algoritmo apriori
- ▶ Reglas de asociación
- ▶ Grocery dataset en kaggle
 - <https://www.kaggle.com/shazadudwadia/supermarket>
- ▶ kaggle:
 - Más de 37 000 datasets en kaggle
 - Registro gratuito mediante Google, etc.
 - Incluye notebooks para trabajar en Python o R online
 - Incluye también cursos

Ejemplo práctico de aplicación (apriori)

- ▶ Grocery dataset en kaggle
 - <https://www.kaggle.com/shazadudwadia/supermarket>
- ▶ En lugar de emplear un notebook, trabajaremos offline
- ▶ Descargamos y descomprimos el .ZIP
 - Download y descomprimos
 - **GroceryStoreDataSet.csv**
 - "MILK,BREAD,BISCUIT"
 - "BREAD,MILK,BISCUIT,CORNFLAKES"
 - "BREAD,TEA,BOURNVITA"
 - "JAM,MAGGI,BREAD,MILK"
 - ...
 - No tiene header alguno
 - Cada línea representa una cesta de la compra de un cliente

Ejemplo práctico de aplicación (apriori)

▶ <http://rasbt.github.io/mlxtend/>

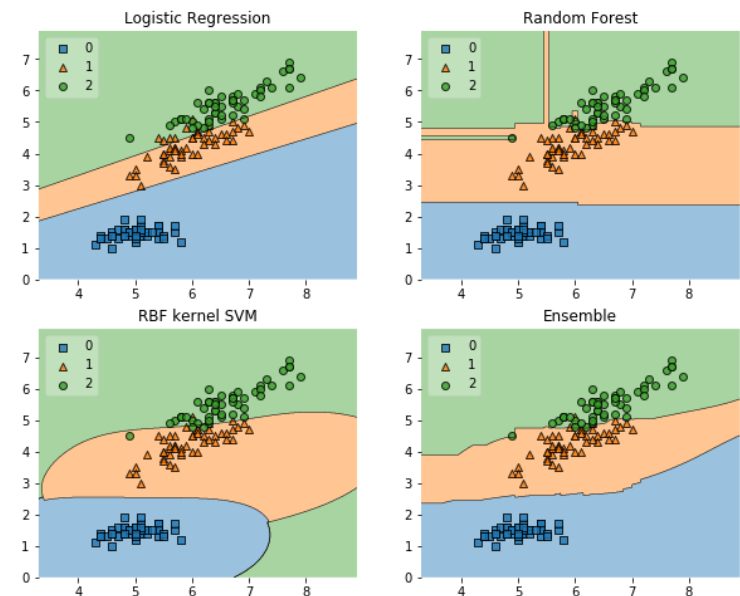
▶ PS C:\Users\xxx> `pip install mlxtend`



▶ **MLxtend** (*Machine Learning extensions*) incluye extensiones útiles para realizar técnicas de machine learning, incluyendo, por ejemplo, el algoritmo *apriori*.

▶ Además, incluye interesantes ayudas para utilizar gráficos como regiones de decisión

— Aunque no lo usaremos, pues ahora no estamos trabajando con clasificadores



Cargamos y visualizamos el dataset

```
# Librerías necesarias
import pandas as pd
# recordad emplear aquí la ruta absoluta o relativa hasta
# nuestro dataset descargado
# en función de en que ruta estamos ejecutando nuestro script
df = pd.read_csv('./GroceryStoreDataSet.csv', names=['products
'], header=None)

print("---df---")
print(df)
print("---df.columns---")
print(df.columns)
print("---df.values---")
print(df.values)
```

Cargamos y visualizamos el dataset

---df---

```

                                products
0                MILK,BREAD,BISCUIT
1    BREAD,MILK,BISCUIT,CORNFLAKES
2                BREAD,TEA,BOURNVITA
3                JAM,MAGGI,BREAD,MILK
4                MAGGI,TEA,BISCUIT
5                BREAD,TEA,BOURNVITA
6                MAGGI,TEA,CORNFLAKES
7    MAGGI,BREAD,TEA,BISCUIT
8                JAM,MAGGI,BREAD,TEA
9                BREAD,MILK
10   COFFEE,COCK,BISCUIT,CORNFLAKES
11   COFFEE,COCK,BISCUIT,CORNFLAKES
12                COFFEE,SUGER,BOURNVITA
13                BREAD,COFFEE,COCK
14                BREAD,SUGER,BISCUIT
15   COFFEE,SUGER,CORNFLAKES
16                BREAD,SUGER,BOURNVITA
17                BREAD,COFFEE,SUGER
18                BREAD,COFFEE,SUGER
19   TEA,MILK,COFFEE,CORNFLAKES
```

Cargamos y visualizamos el dataset

```
---df.columns---
```

```
Index(['products'], dtype='object')
```

```
---df.values---
```

```
[['MILK,BREAD,BISCUIT']  
 ['BREAD,MILK,BISCUIT,CORNFLAKES']  
 ['BREAD,TEA,BOURNVITA']  
 ['JAM,MAGGI,BREAD,MILK']  
 ['MAGGI,TEA,BISCUIT']  
 ['BREAD,TEA,BOURNVITA']  
 ['MAGGI,TEA,CORNFLAKES']  
 ['MAGGI,BREAD,TEA,BISCUIT']  
 ['JAM,MAGGI,BREAD,TEA']  
 ['BREAD,MILK']  
 ['COFFEE,COCK,BISCUIT,CORNFLAKES']  
 ['COFFEE,COCK,BISCUIT,CORNFLAKES']  
 ['COFFEE,SUGER,BOURNVITA']  
 ['BREAD,COFFEE,COCK']  
 ['BREAD,SUGER,BISCUIT']  
 ['COFFEE,SUGER,CORNFLAKES']  
 ['BREAD,SUGER,BOURNVITA']  
 ['BREAD,COFFEE,SUGER']  
 ['BREAD,COFFEE,SUGER']  
 ['TEA,MILK,COFFEE,CORNFLAKES']]
```

Preprocesamos los datos

```
# Librerías necesarias
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
# recordad emplear aquí la ruta absoluta o relativa hasta nuestro da
taset descargado
# en función de en que ruta estamos ejecutando nuestro script
df = pd.read_csv('./GroceryStoreDataSet.csv', names=['products'], he
ader=None)
# preprocesamos los datos usando las comas como separadores
data = list(df["products"].apply(lambda x:x.split(',')))
print("---data---")
print(data)
print()
# convertimos los datos a un formato que entienda mlxtend
te = TransactionEncoder()
te_data = te.fit(data).transform(data)
df = pd.DataFrame(te_data, columns=te.columns_)
print("---df.head()---")
print(df.head())
print()
```

Preprocesamos los datos

---data---

```
[['MILK', 'BREAD', 'BISCUIT'],  
['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],  
['BREAD', 'TEA', 'BOURNVITA'],  
['JAM', 'MAGGI', 'BREAD', 'MILK'],  
['MAGGI', 'TEA', 'BISCUIT'],  
['BREAD', 'TEA', 'BOURNVITA'],  
['MAGGI', 'TEA', 'CORNFLAKES'],  
['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],  
['JAM', 'MAGGI', 'BREAD', 'TEA'],  
['BREAD', 'MILK'],  
['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],  
['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],  
['COFFEE', 'SUGER', 'BOURNVITA'],  
['BREAD', 'COFFEE', 'COCK'],  
['BREAD', 'SUGER', 'BISCUIT'],  
['COFFEE', 'SUGER', 'CORNFLAKES'],  
['BREAD', 'SUGER', 'BOURNVITA'],  
['BREAD', 'COFFEE', 'SUGER'],  
['BREAD', 'COFFEE', 'SUGER'],  
['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

Preprocesamos los datos

---df.head()---

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	SUGER	TEA
0	True	False	True	False	False	False	False	False	True	False	False
1	True	False	True	False	False	True	False	False	True	False	False
2	False	True	True	False	False	False	False	False	False	False	True
3	False	False	True	False	False	False	True	True	True	False	False
4	True	False	False	False	False	False	False	True	False	False	True

Aplicación del algoritmo apriori

Librerías necesarias

```
import pandas as pd
```

```
from mlxtend.preprocessing import TransactionEncoder
```

```
from mlxtend.frequent_patterns import apriori
```

recordad emplear aquí la ruta absoluta o relativa hasta nuestro dataset descargado

en función de en que ruta estamos ejecutando nuestro script

```
df = pd.read_csv('./GroceryStoreDataSet.csv', names=['products'], header=None)
```

preprocesamos los datos usando las comas como separadores

```
data = list(df["products"].apply(lambda x:x.split(',')))
```

convertimos los datos a un formato que entienda mlxtend

```
te = TransactionEncoder()
```

```
te_data = te.fit(data).transform(data)
```

```
df = pd.DataFrame(te_data, columns=te.columns_)
```

procedemos a aplicar el algoritmo apriori

```
df1 = apriori(df, min_support=0.01, use_colnames=True)
```

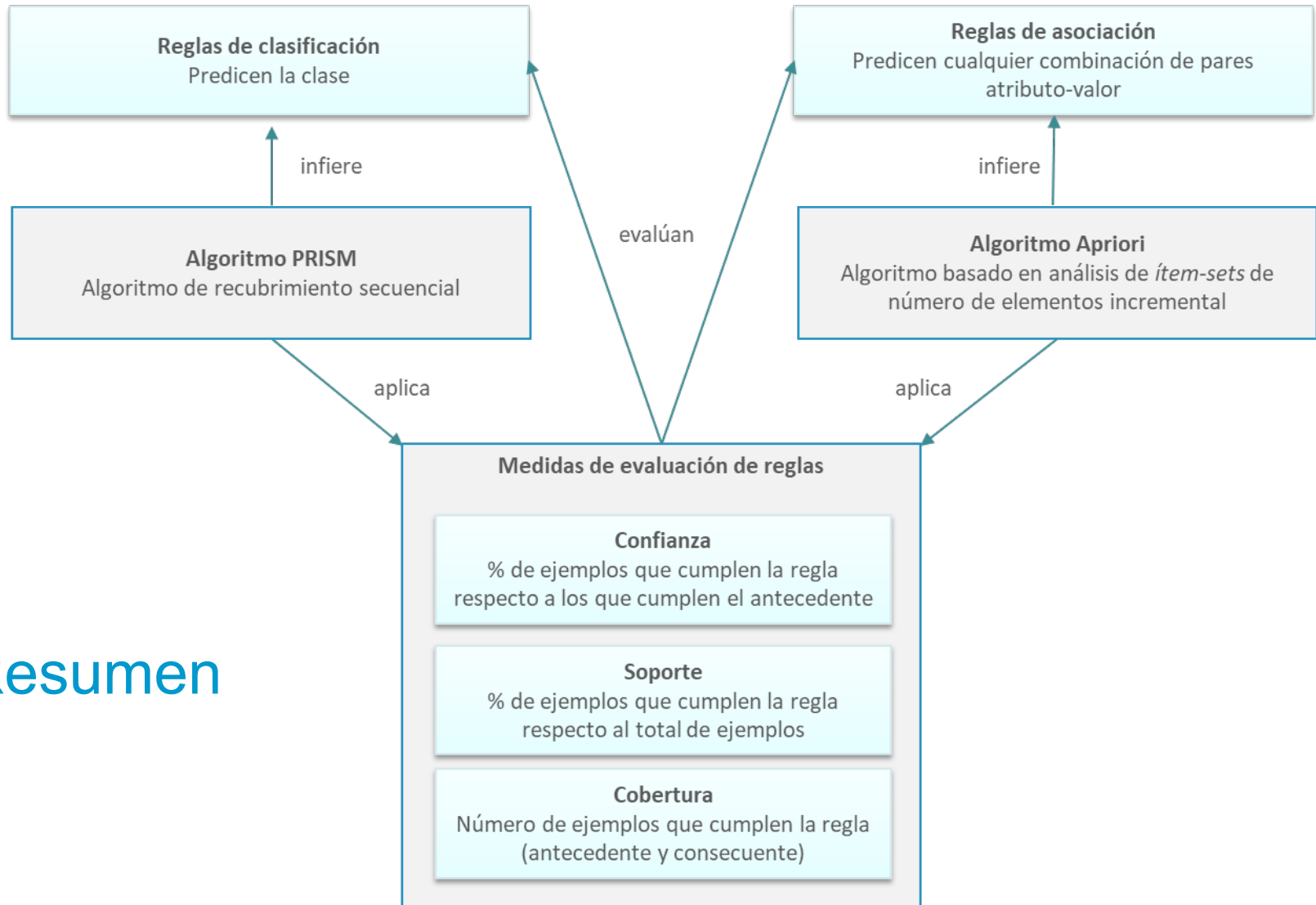
```
df1 = df1.sort_values(by="support", ascending=False)
```

```
print(df1)
```


Obtenemos asociaciones ordenadas de forma descendiente en base a su soporte

	support	itemsets
2	0.65	(BREAD)
4	0.40	(COFFEE)
0	0.35	(BISCUIT)
10	0.35	(TEA)
5	0.30	(CORNFLAKES)
..
55	0.05	(CORNFLAKES, MILK, BISCUIT)
57	0.05	(SUGER, BREAD, BOURNVITA)
17	0.05	(SUGER, BISCUIT)
37	0.05	(CORNFLAKES, MAGGI)
82	0.05	(COFFEE, MILK, TEA, CORNFLAKES)

Reglas



Resumen

Gracias por vuestra atención ¿Dudas?



*Imagen por Peggy und Marco Lachmann-Anke
Licencia: Creative Commons Zero*

UNIVERSIDAD
INTERNACIONAL
DE LA RIOJA

unir

www.unir.net