

Sesión de ejercicios 3

Andrés Auquilla

July 31, 2024

Contents

1	Inducción de Reglas	3
1.1	Inducción de Reglas	3
2	Inducción de Asociación	4
2.1	Algoritmo APRIORI	4

1 Inducción de Reglas

1.1 Inducción de Reglas

Considere el dataset de la Figura 1

ex	temp	hum	wind	play
e_1	mild	norm	strong	pos
e_2	mild	norm	weak	pos
e_3	cool	norm	weak	pos
e_4	mild	high	strong	neg
e_5	cool	norm	strong	neg

Figure 1: Datos de ejemplo para la inducción de reglas

- a. Muestre los literales candidatos para la primera iteración de
 - (a) Top-down example-driven algoritmo LearnOneRule (primera selección e_1)
- b. Encontrar un conjunto de reglas que cubra todos los ejemplos positivos de la Figura 1 usando el algoritmo top-down example-driven LearnOneRule (comenzar siempre por el primer ejemplo positivo que no es cubierto). Como heurística, use accuracy para guiar la búsqueda

En la Figura 2 se presenta el algoritmo "separate-and-conquer" para aprendizaje de reglas. La función *performance()* calcula el porcentaje de aciertos de un literal con respecto al número total de ejemplos cubiertos por dicho literal.

Separate-and-conquer Rule Learning

```
function LearnRuleSet( $E^{\oplus}$ ,  $E^{\ominus}$ ):
    LearnedRules :=  $\emptyset$ 
    while  $E^{\oplus} \neq \emptyset$ , do
        pick  $e$  from  $E^{\oplus}$ 
        Rule := LearnOneRule( $e$ ,  $E^{\oplus}$ ,  $E^{\ominus}$ )
        LearnedRules := LearnedRules  $\cup$  {Rule}
         $E^{\oplus} := E^{\oplus} -$ 
            {examples classified correctly by Rule}
    return LearnedRules

function LearnOneRule( $e$ ,  $E^{\oplus}$ ,  $E^{\ominus}$ ):
    NewRule := "IF true THEN  $\oplus$ "
    NewRuleNeg :=  $E^{\ominus}$ 
    while NewRuleNeg  $\neq \emptyset$ , do
        Candidates := GenerateCandidateLiterals( $e$ )
        BestLit := argmax $_{L \in \text{Candidates}}$ 
            performance(Specialise(NewRule, L),  $E^{\oplus}$ ,  $E^{\ominus}$ )
        NewRule := Specialise(NewRule, BestLit)
        NewRuleNeg := { $x \in E^{\ominus} \mid x$  covered by NewRule}
    return NewRule

function Specialise(Rule, Lit):
    let Rule = "IF conditions THEN  $\oplus$ "
    return "IF conditions  $\wedge$  Lit THEN  $\oplus$ "
```

Figure 2: Algoritmo para aprendizaje de reglas

2 Inducción de Asociación

2.1 Algoritmo APRIORI

Este ejercicio no es obligatorio; por ende, si está complicado con el tiempo, pase al siguiente ejercicio.

Si decide realizar este ejercicio:

- a. Abra el archivo *supermarket.arff* mediante *WEKA*
- b. Vaya a la pestaña *Associate* y seleccione el algoritmo *APRIORI*
- c. Modifique los parámetros: *metricType* y *minMetric* y verifique los resultados
- d. Abra el archivo *apriori.ipynb* y siga las instrucciones