Universidad de Huelva

Tema 3

Clasificación automática

Índice

3.1 Introducción

3.2 Árboles de decisión

3.3 Reglas de clasificación

3.1.1 Definición

- Clasificación supervisada:
  - El conjunto de datos está formado por tuplas atributo-valor
  - El problema de clasificación puede ser biclásico (elegir entre dos clases) o multiclásico (entre muchas clases)
  - Los atributos pueden ser continuos (valores enteros o reales) o discretos (etiquetas)
  - Puede existir ruido (ejemplos mal clasificados)
  - Pueden existir datos incompletos (missing values)

3.1.2 Tipos de clasificadores

- Vecinos más cercanos (K-NN)
- Máquinas de Vectores Soporte (SVM)
- Redes neuronales
- Redes bayesianas
- Árboles de decisión
- Reglas de clasificación
- Sistemas difusos

3.1.3 Criterios de evaluación de clasificadores

- **Matriz de confusión**: muestra la distribución de los errores cometidos por un clasificador a lo largo de las distintas categorías del problema

		Clase verdadera		
		0 (+)	1 (-)	
Clase predicha	0 (+)	a	b	p0
	1 (-)	c	d	p1
		π0	π1	N

- **Tasa de error**: (b+c) / N
- **Sensibilidad**: a/(a+c) proporción de verdaderos positivos
- **Especificidad**: d/(b+d) proporción de verdaderos negativos

3.1.3 Criterios de validación de clasificadores

- **Holdout**: Se divide el conjunto de casos en dos grupos: conjunto de entrenamiento (2/3) y conjunto de test (1/3). El conjunto de entrenamiento se usa para generar el clasificador y el de test para evaluarlo.
- **Validación cruzada** (**cross-validation**): Se divide el conjunto de casos en K subconjuntos del mismo tamaño. Se utilizan K-1 subconjuntos como datos de entrenamiento y 1 subconjunto como datos de test. Se repite para los K subconjuntos y se calcula la media de la evaluación. Suele utilizarse K=10.
- **Dejar uno fuera** (**leave one out**): validación cruzada con K igual al número de casos.
- **Bootstrapping**: el conjunto de entrenamiento se escoge como una muestra aleatoria con reemplazamiento.

3.2.1 Definición

- **Árbol de decisión:**
  - es una representación de los procesos de decisión involucrados en las tareas de clasificación
- **Elementos**:
  - Hojas: describen la etiqueta asociada a una clasificación
  - Nodos: describen la pregunta acerca de un cierto atributo
  - Ramas de un nodo: representan los diferentes valores que puede tomar el atributo respecto del que se pregunta en el nodo
- **Objetivos**:
  - Dado un conjunto de ejemplos clasificados, generar el árbol de decisión óptimo, es decir, aquel que permita describir la clasificación con el menor número de cuestiones posible

3.2.2 Un poco de historia

- En 1966 se publica "*Experiments in Induction*", de Hunt, Marin y Stone donde describen lo que los autores denominan *Concept Learning Systems* (CLS), que utilizan atributos binarios y técnicas heurísticas para construir árboles de decisión.
- En 1984 se publica "*Classification and regression trees*", de Breiman, Friedman, Olshen y Stone. En él se describe un método de inducción para construir árboles de decisión de forma recursiva que se conoce como CART.
- En 1986 J. Ross Quinlan desarrolla ID3 (*Iterative Dichotomiser 3*), que posteriormente mejoraría creando C4.5 (1993). ID3 utiliza la entropía de información para crear los árboles.

3.2.2 Un poco de historia

- Otro algoritmo muy extendido para la creación de árboles de decisión es CHAID, que se incluye en muchos paquetes estadísticos. CHAID utiliza el contraste de Pearson (o test χ^2) para seleccionar el atributo a estudiar en cada nodo.
- Otros algoritmos menos conocidos son los siguientes
  - ID4 e ID4R de Schlimmer y Fisher (1986): es una versión incremental de ID3
  - ID5 e ID5R de Utgoff (1990): versión incremental de ID3
  - J4.8: implementación de C4.5 incluida en WEKA
  - C5.0: última versión mejorada de C4.5

3.2.3 El algoritmo ID3

ID3(Instancias)

SI todas las instancias son de la misma clase C ENTONCES
devolver Hoja(C)

SINO SI el conjunto de instancias está vacío ENTONCES devolver Hoja(Clase_por_defecto)

SINO SI el conjunto de instancias no contiene ningún atributo ENTONCES devolver Hoja(Clase_mayoritaria)

SINO

- Elegir atributo A con mayor ganancia de información
- Crear nodo con el atributo seleccionado
- Para cada valor V del atributo A
  - Crear una rama con el valor V
  - Seleccionar las instancias con el valor V del atributo A
  - Eliminar el atributo A de este conjunto de instancias Cv
  - Asignar a la rama el árbol devuelto por ID3(Cv)
- Devolver nodo

3.2.3 El algoritmo ID3

- Definiciones:
  - **C**: conjunto de clases
  - $\mathbf{A_i}$: conjunto de atributos
  - $\mathbf{V_i}$: conjunto de valores de $\mathbf{A_i}$
  - **n**: número de patrones
  - $\mathbf{n_c}$: número de patrones de la clase $\mathbf{c} \in \mathbf{C}$
  - $\mathbf{n_{ij}}$: número de patrones con el valor $\mathbf{j} \in \mathbf{V_i}$ en el atributo $\mathbf{A_i}$
  - $\mathbf{n_{ijc}}$: número de patrones de la clase **c** con el valor **j** en el atributo $\mathbf{A_i}$

3.2.3 El algoritmo ID3

- Ganancia de información del atributo $\mathbf{A_i}$
  - $\mathbf{G(A_i) = I - I(A_i)}$
- Entropía de información del conjunto de patrones
  - $\mathbf{I} = - \sum (n_c/n) \log_2(n_c /n)$
- Entropía de información del atributo $\mathbf{A_i}$
  - $\mathbf{I(A_i)} = \sum (n_{ij}/n) I_{ij}$
- Entropía de información del valor j del atributo $\mathbf{A_i}$
  - $\mathbf{I_{ij}} = - \sum (n_{ijc}/ n_{ij}) \log_2(n_{ijc}/ n_{ij})$

3.2.3 El algoritmo ID3

- Ejemplo:
  - Se desea generar un árbol de decisión que clasifique entre células normales y células cancerígenas según los datos de la siguiente tabla:

Ejemplo	Antenas	Colas	Núcleos	Cuerpo	Clase
1	1	0	2	Rayado	Normal
2	1	0	1	Blanco	Cancerígena
3	1	2	0	Rayado	Normal
4	0	2	1	Rayado	Normal
5	1	1	1	Rayado	Cancerígena
6	2	2	1	Rayado	Cancerígena

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Antenas**
  - $\mathbf{I(A_1)} = \sum (n_{ij}/n)\, I_{ij} = (1/6) \cdot I_{10} + (4/6) \cdot I_{11} + (1/6) \cdot I_{12} = 0.66$
- Entropía de información del valor 0 del atributo **Antenas**
  - $\mathbf{I_{10}} = - (1/1) \log_2(1/1) - (0/1) \log_2 (0/1) = 0$
- Entropía de información del valor 1 del atributo **Antenas**
  - $\mathbf{I_{11}} = - (2/4) \log_2 (2/4) - (2/4) \log_2 (2/4) = 1$
- Entropía de información del valor 2 del atributo **Antenas**
  - $\mathbf{I_{12}} = - (0/1) \log_2(0/1) - (1/1) \log_2 (1/1) = 0$

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Colas**
  - $\mathbf{I(A_2)} = \sum (n_{ij}/n)\, I_{ij} = (2/6) \cdot I_{20} + (1/6) \cdot I_{21} + (3/6) \cdot I_{22} = 0.79$
- Entropía de información del valor 0 del atributo **Colas**
  - $\mathbf{I_{20}} = - (1/2) \log_2(1/2) - (1/2) \log_2 (1/2) = 1$
- Entropía de información del valor 1 del atributo **Colas**
  - $\mathbf{I_{21}} = - (0/1) \log_2 (0/1) - (1/1) \log_2 (1/1) = 0$
- Entropía de información del valor 2 del atributo **Colas**
  - $\mathbf{I_{22}} = - (2/3) \log_2(2/3) - (1/3) \log_2 (1/3) = 0.9183$

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Núcleos**
  - $\mathbf{I(A_3)} = \sum (n_{ij}/n)\, I_{ij} = (1/6) \cdot I_{30} + (4/6) \cdot I_{31} + (1/6) \cdot I_{32} = 0.54$
- Entropía de información del valor 0 del atributo **Núcleos**
  - $\mathbf{I_{30}} = - (1/1) \log_2(1/1) - (0/1) \log_2 (0/1) = 0$
- Entropía de información del valor 1 del atributo **Núcleos**
  - $\mathbf{I_{31}} = - (1/4) \log_2 (1/4) - (3/4) \log_2 (3/4) = 0.8113$
- Entropía de información del valor 2 del atributo **Núcleos**
  - $\mathbf{I_{32}} = - (1/1) \log_2(1/1) - (0/1) \log_2 (0/1) = 0$

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Cuerpo**
  - $\mathbf{I(A_4)} = \sum (n_{ij}/n)\, I_{ij} = (1/6) \cdot I_{40} + (5/6) \cdot I_{41} = 0.81$
- Entropía de información del valor blanco del atributo **Cuerpo**
  - $\mathbf{I_{40}} = -\,(0/1)\log_2(0/1) - (1/1)\log_2(1/1) = 0$
- Entropía de información del valor rayado del atributo **Cuerpo**
  - $\mathbf{I_{41}} = -\,(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.9710$

3.2.3 El algoritmo ID3

- Se escoge el atributo Núcleos
- Se dividen los patrones:
  - Núcleos = 0

Ejemplo	Antenas	Colas	Cuerpo	Clase
3	1	2	Rayado	Normal

  - Núcleos = 1

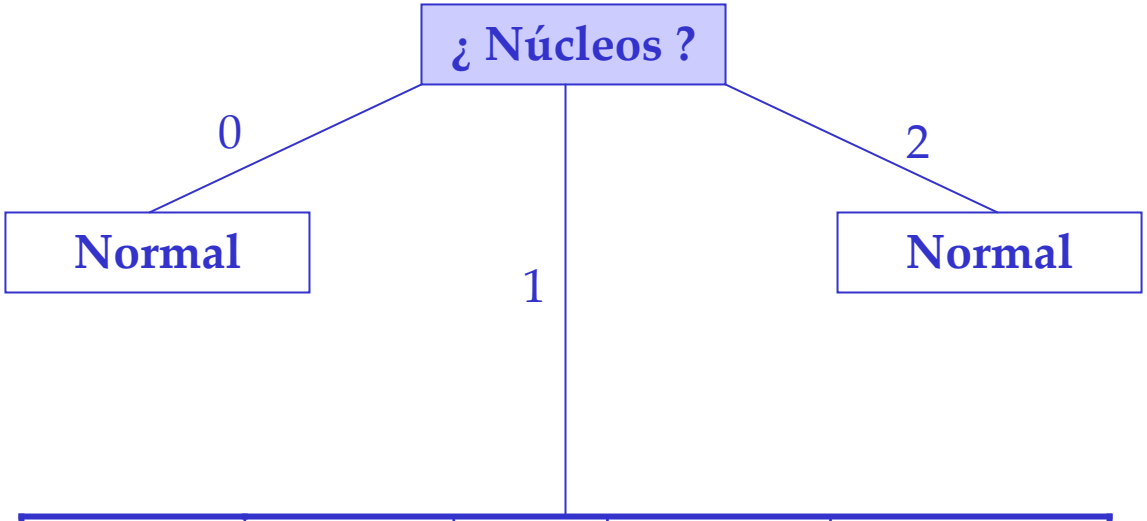
Ejemplo	Antenas	Colas	Cuerpo	Clase
2	1	0	Blanco	Cancerígena
4	0	2	Rayado	Normal
5	1	1	Rayado	Cancerígena
6	2	2	Rayado	Cancerígena

  - Núcleos = 2:

Ejemplo	Antenas	Colas	Cuerpo	Clase
1	1	0	Rayado	Normal

3.2.3 El algoritmo ID3

- Árbol de decisión



Ejemplo	Antenas	Colas	Cuerpo	Clase
2	1	0	Blanco	Cancerígena
4	0	2	Rayado	Normal
5	1	1	Rayado	Cancerígena
6	2	2	Rayado	Cancerígena

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Antenas**
  - $\mathbf{I(A_1)} = \sum (n_{ij}/n)\, I_{ij} = (1/4) \cdot I_{10} + (2/4) \cdot I_{11} + (1/4) \cdot I_{12} = 0$
- Entropía de información del valor 0 del atributo **Antenas**
  - $\mathbf{I_{10}} = -\,(1/1)\log_2(1/1) - (0/1)\log_2(0/1) = 0$
- Entropía de información del valor 1 del atributo **Antenas**
  - $\mathbf{I_{11}} = -\,(0/2)\log_2(0/2) - (2/2)\log_2(2/2) = 0$
- Entropía de información del valor 2 del atributo **Antenas**
  - $\mathbf{I_{12}} = -\,(0/1)\log_2(0/1) - (1/1)\log_2(1/1) = 0$

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Colas**
  - $\mathbf{I(A_2)} = \sum (n_{ij}/n)\, I_{ij} = (1/4) \cdot I_{20} + (1/4) \cdot I_{21} + (2/4) \cdot I_{22} = 0.5$
- Entropía de información del valor 0 del atributo **Colas**
  - $\mathbf{I_{20}} = - (0/1) \log_2(0/1) - (1/1) \log_2 (1/1) = 0$
- Entropía de información del valor 1 del atributo **Colas**
  - $\mathbf{I_{21}} = - (0/1) \log_2 (0/1) - (1/1) \log_2 (1/1) = 0$
- Entropía de información del valor 2 del atributo **Colas**
  - $\mathbf{I_{22}} = - (1/2) \log_2(1/2) - (1/2) \log_2 (1/2) = 1$

3.2.3 El algoritmo ID3

- Entropía de información del atributo **Cuerpo**
  - $\mathbf{I(A_4)} = \sum (n_{ij}/n)\, I_{ij} = (1/4) \cdot I_{40} + (3/4) \cdot I_{41} = 0.6887$
- Entropía de información del valor blanco del atributo **Cuerpo**
  - $\mathbf{I_{40}} = - (0/1) \log_2(0/1) - (1/1) \log_2 (1/1) = 0$
- Entropía de información del valor rayado del atributo **Cuerpo**
  - $\mathbf{I_{41}} = - (1/3) \log_2 (1/3) - (2/3) \log_2 (2/3) = 0.9183$

3.2.3 El algoritmo ID3

- Se escoge el atributo Antenas
- Se dividen los patrones:
  - Antenas = 0

Ejemplo	Colas	Cuerpo	Clase
4	2	Rayado	Normal

  - Antenas = 1

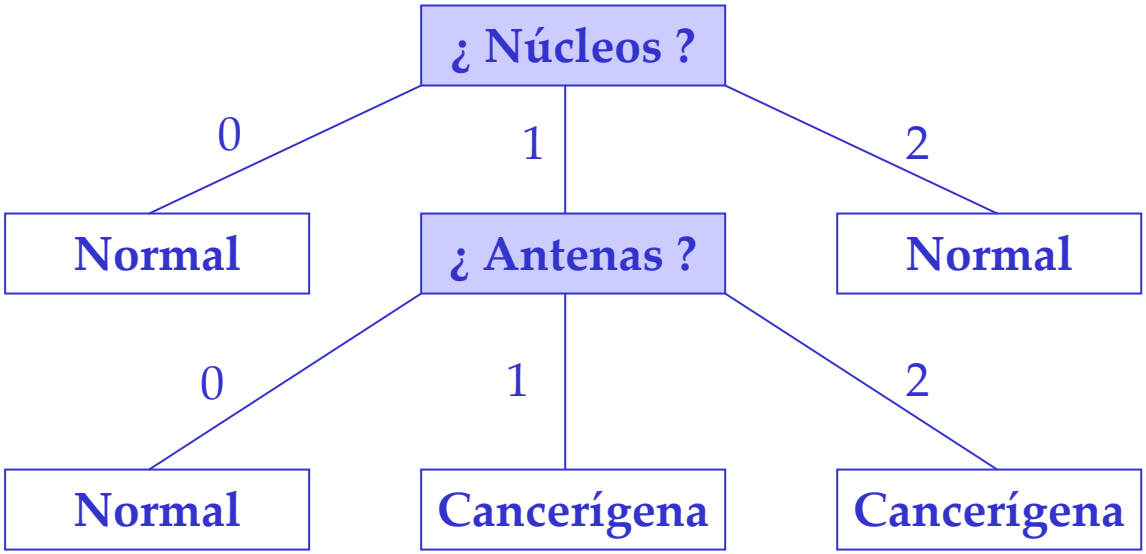
Ejemplo	Colas	Cuerpo	Clase
2	0	Blanco	Cancerígena
5	1	Rayado	Cancerígena

  - Antenas = 2:

Ejemplo	Colas	Cuerpo	Clase
6	2	Rayado	Cancerígena

3.2.3 El algoritmo ID3

- Árbol de decisión



3.3.1 El algoritmo AQ

- **Origen**: Michalski (1983) desarrolla una metodología genérica denominada *star* de la que derivan numerosos algoritmos de generación de reglas que se agrupan bajo la denominación Aq.
- **Objetivos**: Dado un conjunto de ejemplos positivos y negativos, generar un conjunto de reglas que describan todos los positivos y no reconozcan ningún ejemplo negativo.
- **Elementos**:
  - *A*, conjunto de atributos
  - *V*, conjunto de valores de los atributos
  - *C*, conjunto de clases
  - *E*, conjunto de ejemplos de entrenamiento
  - *LEF*, lista de criterios de preferencia de reglas

3.3.1 El algoritmo AQ

- **LEF:** función de evaluación lexicográfica
  - Permite elegir la regla a añadir entre un conjunto de candidatas
  - Posibles criterios:
    - Cobertura: número de ejemplos positivos cubiertos
    - Simplicidad: número de atributos que se estudian en el antecedente
    - Coste: coste de evaluación del antecedente
    - Generalidad: número de ejemplos observados entre el número de ejemplos posibles
  - El algoritmo Aq es independiente de la función de evaluación

3.3.1 El algoritmo AQ

- **Selector**: permite realizar pregunta sobre un atributo:
  - **Sintaxis**: (*Atributo Operador Valores*)
  - **Operador**: =, <=, >, !=
  - **Valores**: valor continuo o discreto del atributo
- **Complejo**: es una conjunción de selectores
  - Ejemplo: (Peso > 30) ^ (Color = rojo)
  - Permite describir el antecedente de una regla de clasificación
  - Equivale a: SI (Peso > 30) ^ (Color = rojo) ENTONCES clase = positivo
- **Recubrimiento**: es una disyunción de complejos
  - Ejemplo: [(Peso>30) ^ (Color = rojo)] v [(Tamaño = grande)]
  - Permite describir conjuntos de reglas

3.3.1 El algoritmo AQ

- **Algoritmo para generar lar reglas:**
  - Inicialmente el conjunto de reglas (recubrimiento) esta vacío
  - Se considera el conjunto P de ejemplos positivos y el conjunto N de ejemplos negativos
  - Mientras queden ejemplos positivos en P, repetir
    - Elegir un ejemplo de P que será la semilla de la próxima regla
    - Generar complejos que cubran la semilla y excluyan a los ejemplos de N (algoritmo *star*)
    - Elegir de entre los complejos el que optimice el criterio de selección (LEF)
    - Añadir el complejo elegido al recubrimiento
    - Eliminar de P todos los ejemplos cubiertos por la nueva regla

3.3.1 El algoritmo AQ

- **Algoritmo *star* :**
  - Sea E el conjunto de complejos a devolver:
    - Inicialmente: E = conjunto vacío
  - Sea L una lista de complejos a estudiar:
    - Inicialmente: L = { () }, es decir contiene un complejo que acepta todo
  - Sea S el conjunto de selectores de la semilla
    - S se forma a con los atributos y valores del ejemplo semilla
  - Mientras L no esté vacía repetir
    - Crear un conjunto E' con complejos creados por conjunción de un elemento de L y un selector de S
    - Eliminar de E' los elementos que ya estén incluidos en E
    - Para cada complejo de E', si no cubre ningún ejemplo negativo, entonces
      - Añadir el complejo a E
      - Eliminar el complejo de E'
    - Actualizar la lista L a los elementos de E'
  - Devolver el conjunto E

3.3.1 El algoritmo AQ

- EJEMPLO: Se desea generar un conjunto de reglas de clasificación que distinga entre células normales y células cancerígenas según los datos de las siguientes tablas:

Normales

Antenas	Colas	Núcleos	Cuerpo
1	0	2	Rayado
1	2	0	Rayado
0	2	1	Rayado
0	2	2	Rayado

Cancerígenas

Antenas	Colas	Núcleos	Cuerpo
1	0	1	Blanco
1	1	1	Rayado
2	2	1	Rayado

3.3.1 El algoritmo AQ

EJEMPLO

- Inicialmente el conjunto de reglas está vacío
- Se escoge como semilla el primer ejemplo positivo
  - (Antenas = 1) ^ (Colas = 0) ^ (Núcleos = 2) ^ (Cuerpo = Rayado)
- Se genera el conjunto de selectores S
  - S = {(Antenas = 1) , (Colas = 0) , (Núcleos = 2) , (Cuerpo = Rayado) }

PRIMERA ITERACIÓN

- Se crea el conjunto E'
  - E' = {(Antenas = 1) , (Colas = 0) , (Núcleos = 2) , (Cuerpo = Rayado) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Núcleos = 2) }
- Se almacena en L los elementos restantes
  - L = {(Antenas = 1) , (Colas = 0) , (Cuerpo = Rayado) }

3.3.1 El algoritmo AQ

SEGUNDA ITERACIÓN

- Se crea el conjunto E'
  - E' = { (Antenas = 1)^(Colas = 0) , (Antenas = 1)^(Núcleos = 2) ,
  - (Antenas = 1)^(Cuerpo = Rayado) , (Colas = 0)^(Núcleos = 2) ,
  - (Colas = 0)^(Cuerpo = Rayado) , (Núcleos = 2)^(Cuerpo = Rayado) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Núcleos = 2) , (Antenas = 1)^(Núcleos = 2) , (Colas = 0)^(Núcleos = 2) ,
  - (Colas = 0)^(Cuerpo = Rayado), (Núcleos = 2)^(Cuerpo = Rayado) }
- Se almacena en L los elementos restantes
  - L = { (Antenas = 1)^(Colas = 0) , (Antenas = 1)^(Cuerpo = Rayado) }

3.3.1 El algoritmo AQ

TERCERA ITERACIÓN

- Se crea el conjunto E'
  - E' = { (Antenas = 1)^(Colas = 0)^(Núcleos = 2) ,
  - (Antenas = 1)^(Colas = 0)^(Cuerpo = Rayado) ,
  - (Antenas = 1)^(Cuerpo = Rayado)^(Núcleos = 2) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Núcleos = 2) ,
  - (Antenas = 1)^(Núcleos = 2) ,
  - (Colas = 0)^(Núcleos = 2) ,
  - (Colas = 0)^(Cuerpo = Rayado),
  - (Núcleos = 2)^(Cuerpo = Rayado),
  - (Antenas = 1)^(Colas = 0)^(Núcleos = 2) ,
  - (Antenas = 1)^(Colas = 0)^(Cuerpo = Rayado) ,
  - (Antenas = 1)^(Cuerpo = Rayado)^(Núcleos = 2) }
- Se almacena en L los elementos restantes
  - L = { }

3.3.1 El algoritmo AQ

- Se selecciona la regla con mejor criterio de selección:
  - LEF = { (cobertura = 1), (número de premisas = 3) }
  - Las reglas con máxima cobertura son:
    - (Núcleos = 2)
    - (Núcleos = 2)^(Cuerpo = Rayado)
  - Entre estas la regla con menor número de premisas es:
    - (Núcleos = 2)
  - Se añade a la lista de reglas y se eliminan los ejemplos cubiertos
- Se busca una nueva regla
- Se escoge como semilla el siguiente ejemplo positivo
  - (Antenas = 1) ^ (Colas = 2) ^ (Núcleos = 0) ^ (Cuerpo = Rayado)

3.3.1 El algoritmo AQ

- Se genera el conjunto de selectores S
  - S = {(Antenas = 1) , (Colas = 2) , (Núcleos = 0) , (Cuerpo = Rayado) }

PRIMERA ITERACIÓN

- Se crea el conjunto E'
  - E' = {(Antenas = 1) , (Colas = 2) , (Núcleos = 0) , (Cuerpo = Rayado) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Núcleos = 0) }
- Se almacena en L los elementos restantes
  - L = {(Antenas = 1) , (Colas = 2) , (Cuerpo = Rayado) }

3.3.1 El algoritmo AQ

SEGUNDA ITERACIÓN

- Se crea el conjunto E'
  - E' = { (Antenas = 1)^(Colas = 2) , (Antenas = 1)^(Núcleos = 0) ,
  - (Antenas = 1)^(Cuerpo = Rayado) , (Colas = 2)^(Núcleos = 0) ,
  - (Colas = 2)^(Cuerpo = Rayado) , (Núcleos = 0)^(Cuerpo = Rayado) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Núcleos = 0) ,
  - (Antenas = 1)^(Colas = 2) ,
  - (Antenas = 1)^(Núcleos = 0) ,
  - (Colas = 2)^(Núcleos = 0),
  - (Núcleos = 0)^(Cuerpo = Rayado) }
- Se almacena en L los elementos restantes
  - L = { (Antenas = 1)^(Cuerpo = Rayado) ,
  - (Colas = 2)^(Cuerpo = Rayado) }

3.3.1 El algoritmo AQ

TERCERA ITERACIÓN

- Se crea el conjunto E'
  - E' = { (Antenas = 1)^(Cuerpo = Rayado)^(Colas = 2) ,
  - (Antenas = 1)^(Cuerpo = Rayado)^(Núcleos = 0) ,
  - (Colas = 2)^(Cuerpo = Rayado)^(Núcleos = 0) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Núcleos = 0) ,
  - (Antenas = 1)^(Colas = 2) ,
  - (Antenas = 1)^(Núcleos = 0) ,
  - (Colas = 2)^(Núcleos = 0),
  - (Núcleos = 0)^(Cuerpo = Rayado),
  - (Antenas = 1)^(Cuerpo = Rayado)^(Colas = 2) ,
  - (Antenas = 1)^(Cuerpo = Rayado)^(Núcleos = 0) ,
  - (Colas = 2)^(Cuerpo = Rayado)^(Núcleos = 0) }
- Se almacena en L los elementos restantes
  - L = { }

3.3.1 El algoritmo AQ

- Se selecciona la regla con mejor criterio de selección:
  - LEF = { (cobertura = 1), (número de premisas = 3) }
  - Todas las reglas tienen cobertura 1
  - Entre estas, la regla con menor número de premisas es:
    - (Núcleos = 0)
  - Se añade a la lista de reglas y se eliminan los ejemplos cubiertos
- Reglas actuales:
  - SI (Núcleos = 2) ENTONCES Clase = Normal
  - SI (Núcleos = 0) ENTONCES Clase = Normal
- Se busca una nueva regla
- Se escoge como semilla el último ejemplo positivo
  - (Antenas = 0) ^ (Colas = 2) ^ (Núcleos = 1) ^ (Cuerpo = Rayado)

3.3.1 El algoritmo AQ

- Se genera el conjunto de selectores S
  - S = {(Antenas = 0) , (Colas = 2) , (Núcleos = 1) , (Cuerpo = Rayado) }

PRIMERA ITERACIÓN

- Se crea el conjunto E'
  - E' = {(Antenas = 0) , (Colas = 2) , (Núcleos = 1) , (Cuerpo = Rayado) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Antenas = 0) }
- Se almacena en L los elementos restantes
  - L = { (Colas = 2) , (Núcleos = 1), (Cuerpo = Rayado) }

3.3.1 El algoritmo AQ

SEGUNDA ITERACIÓN

- Se crea el conjunto E'
  - E' = { (Colas = 2)^(Antenas=0) , (Colas = 2)^(Núcleos = 1) ,
  - (Colas = 2)^(Cuerpo = Rayado) , (Núcleos = 1)^(Antenas = 0) ,
  - (Núcleos = 1)^(Cuerpo = Rayado) , (Cuerpo = Rayado)^(Antenas = 0) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Antenas = 0) ,
  - (Colas = 2)^(Antenas=0),
  - (Núcleos = 1)^(Antenas = 0),
  - (Cuerpo = Rayado)^(Antenas = 0)}
- Se almacena en L los elementos restantes
  - L = { (Colas = 2)^(Núcleos = 1) ,
  - (Colas = 2)^(Cuerpo = Rayado),
  - (Núcleos = 1)^(Cuerpo = Rayado)}

3.3.1 El algoritmo AQ

TERCERA ITERACIÓN

- Se crea el conjunto E'
  - L = { (Colas = 2)^(Núcleos = 1)^(Antenas = 0) ,
  - (Colas = 2)^(Núcleos = 1)^(Cuerpo = Rayado) ,
  - (Colas = 2)^(Cuerpo = Rayado)^(Antenas = 0),
  - (Núcleos = 1)^(Cuerpo = Rayado)^(Antenas = 0)}
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Antenas = 0) ,
  - (Colas = 2)^(Antenas=0),
  - (Núcleos = 1)^(Antenas = 0),
  - (Cuerpo = Rayado)^(Antenas = 0),
  - (Colas = 2)^(Núcleos = 1)^(Antenas = 0) ,
  - (Colas = 2)^(Cuerpo = Rayado)^(Antenas = 0),
  - (Núcleos = 1)^(Cuerpo = Rayado)^(Antenas = 0)}
- Se almacena en L los elementos restantes
  - L = {(Colas = 2)^(Núcleos = 1)^(Cuerpo = Rayado) }

3.3.1 El algoritmo AQ

CUARTA ITERACIÓN

- Se crea el conjunto E'
  - L = { (Antenas = 0)^(Colas = 2)^(Núcleos = 1)^(Cuerpo = Rayado) }
- Se almacena en E los elementos que no cubren ejemplos negativos
  - E = { (Antenas = 0) ,
  - (Colas = 2)^(Antenas=0),
  - (Núcleos = 1)^(Antenas = 0),
  - (Cuerpo = Rayado)^(Antenas = 0),
  - (Colas = 2)^(Núcleos = 1)^(Antenas = 0) ,
  - (Colas = 2)^(Cuerpo = Rayado)^(Antenas = 0),
  - (Núcleos = 1)^(Cuerpo = Rayado)^(Antenas = 0),
  - (Antenas = 0)^(Colas = 2)^(Núcleos = 1)^(Cuerpo = Rayado)}
- Se almacena en L los elementos restantes
  - L = { }

3.3.1 El algoritmo AQ

- Se selecciona la regla con mejor criterio de selección:
  - LEF = { (cobertura = 1), (número de premisas = 3) }
  - Todas las reglas tienen cobertura 1
  - Entre estas, la regla con menor número de premisas es:
    - (Antenas = 0)
  - Se añade a la lista de reglas y se eliminan los ejemplos cubiertos
- Reglas actuales:
  - SI (Núcleos = 2) ENTONCES Clase = Normal
  - SI (Núcleos = 0) ENTONCES Clase = Normal
  - SI (Antenas = 0) ENTONCES Clase = Normal
- Ya no quedan ejemplos positivos

3.3.2 El algoritmo CN2

- Es muy parecido al algoritmo AQ. Genera una lista ordenada de reglas (el orden de las reglas de CN2 influye en la clasificación) siendo la última regla la regla por defecto. Para la construcción de las reglas se utilizan todos los selectores posibles (no sólo los de la semilla) y se seleccionan atendiendo a la entropía de distribución y a la significancia estadística de los complejos estudiados

3.3.3 Otros algoritmos

- **Otros algoritmos de creación de reglas de clasificación:**
  - REP (*Reduced Error Pruning*)
  - IREP (*Incremental Reduced Error Pruning*)
  - IREP*
  - RIPPER
  - SLIPPER