Machine Learning

Preprocesamiento:
Reducción de Datos - Discretización

Dr. Edgar Acuña
Departamento de Ciencias Matemáticas
Universidad de Puerto Rico-Mayaguez

E-mail: edgar.acuna@upr.edu, eacunaf@gmail.com

Website: academic.uprm.edu/eacuna

Discretización

Proceso que transforma datos cuantitativos en datos cualitativos.

- Algunos algoritmos de clasificación aceptan solo atributos categóricos (LVF, FINCO, Naïve Bayes, Rough Sets).
- El proceso de aprendizaje frecuentemente es menos eficiente y menos efectivo cuando los datos tienen solamente variables cuantitativas.

Ejemplo de discretizacion aplicado a una parte de Bupa

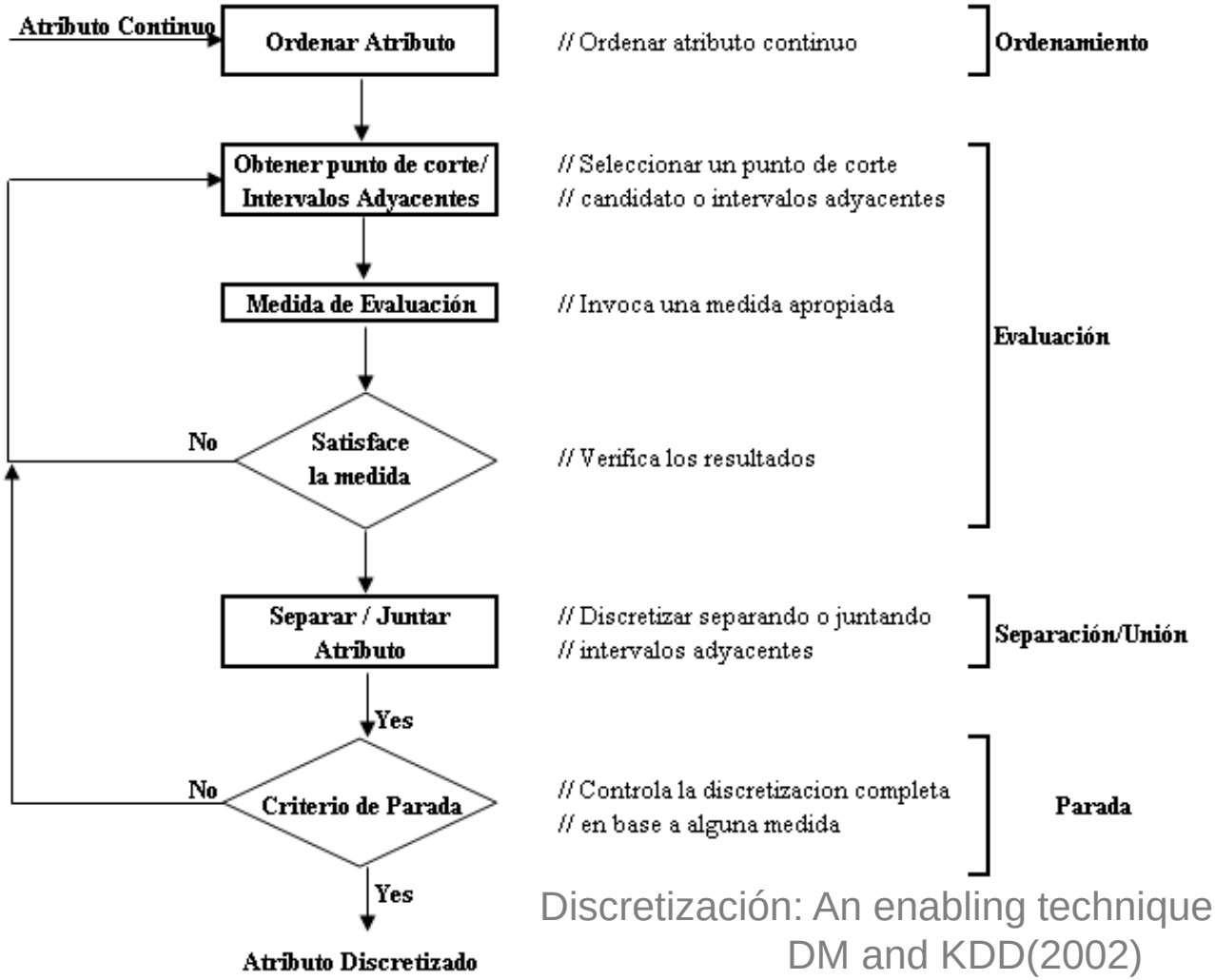
```
> m
   V1  V2  V3  V4 V5
45 5.1 3.8 1.9 0.4  1
46 4.8 3.0 1.4 0.3  1
47 5.1 3.8 1.6 0.2  1
48 4.6 3.2 1.4 0.2  1
49 5.3 3.7 1.5 0.2  1
50 5.0 3.3 1.4 0.2  1
51 7.0 3.2 4.7 1.4  2
52 6.4 3.2 4.5 1.5  2
53 6.9 3.1 4.9 1.5  2
54 5.5 2.3 4.0 1.3  2
55 6.5 2.8 4.6 1.5  2
```

```
> disc.ew(m,1:4)
   V1 V2 V3 V4 V5
45  1  3  1  1  1
46  1  2  1  1  1
47  1  3  1  1  1
48  1  2  1  1  1
49  1  3  1  1  1
50  1  2  1  1  1
51  2  2  2  2  2
52  2  2  2  2  2
53  2  2  2  2  2
54  1  1  2  2  2
55  2  2  2  2  2
```

Top-down (Separar) versus Bottom-up (Juntar)

- Los métodos Top-down inician con una lista vacía de puntos de corte (o split-points) y continúan agregando nuevos puntos a la lista ”separando” los intervalos mientras la discretización progresa.
- Los métodos Bottom-up inician con la lista completa de todos los valores continuos de la variable como puntos de corte y eliminan algunos de ellos “juntando” los intervalos mientras la discretización progresa.

Discretización



Discretización: An enabling technique. Liu et al. DM and KDD(2002)

Discretización Estática vs. Dinámica

- Discretización Dinámica: algunos algoritmos de clasificación tienen incorporados mecanismos para discretizar atributos continuos (por ejemplo, decision trees: CART, C4.5). Los atributos continuos son discretizados durante el proceso de clasificación.
- Discretización Estática: Es un paso más en el preprocesamiento de datos. Los atributos continuos son previamente discretizados antes de la tarea de clasificación.
- No existe una ventaja clara de algunos de los métodos (Dougherty, Kohavi, and Sahami, 1995).

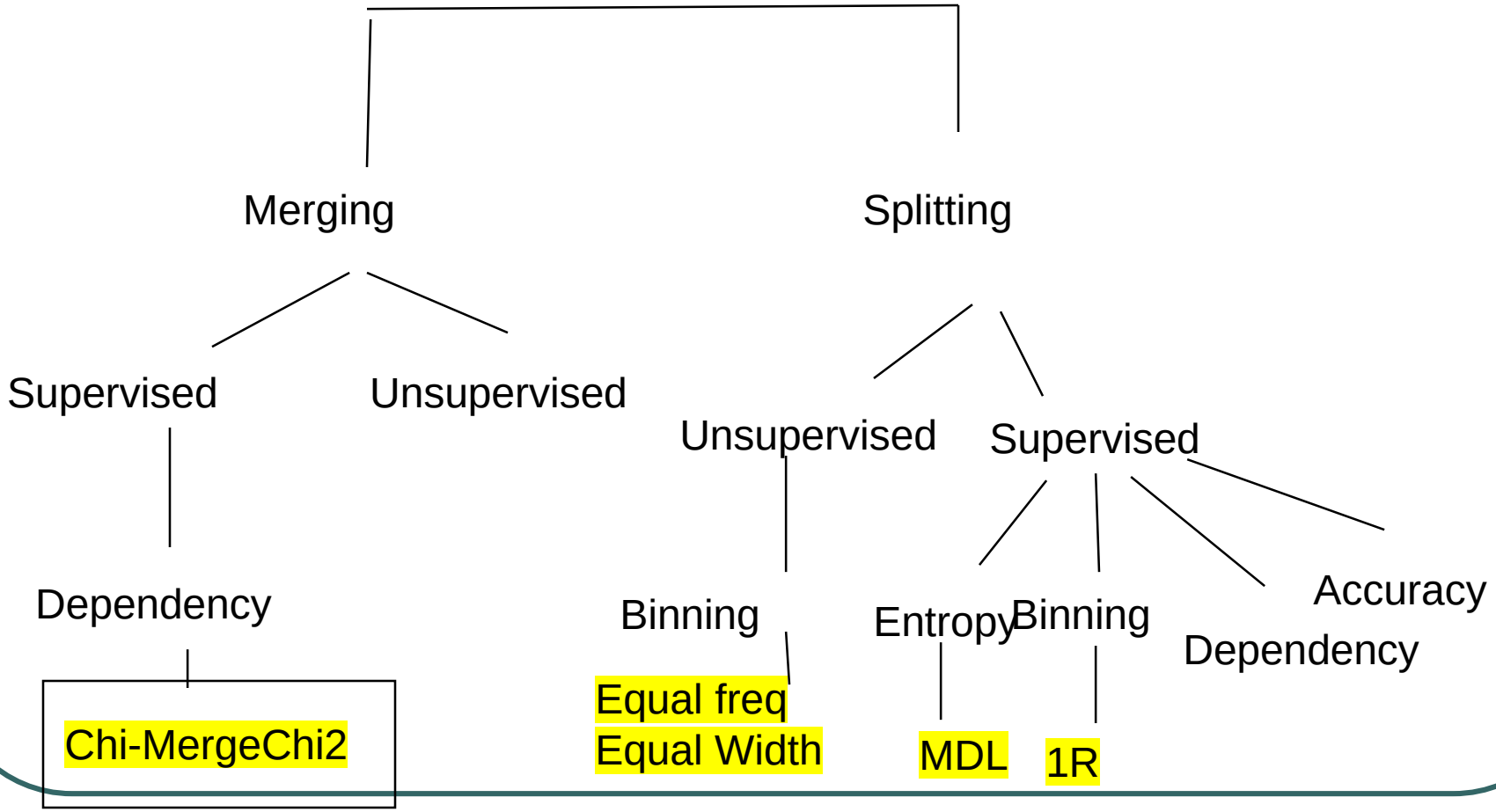
Supervisado versus No supervisado

- Los métodos **supervisados** solo son aplicables cuando se trabaja con datos que están divididos en clases. Estos métodos utilizan la información de la clase cuando se selecciona los puntos de corte en la discretización.
- Los métodos **no supervisados** no utilizan la información de la clase.
- Los métodos supervisados pueden además ser caracterizados como *basados en la tasa de error*, *basados en entropía* o *basados en estadísticas*. Los métodos basados en error aplican un clasificador a los datos transformados y seleccionan los intervalos que minimizan el error en el conjunto de entrenamiento. En contraste, los métodos basados en entropía y los basados en estadísticas evalúan respectivamente la entropía de la clase o alguna otra estadística con respecto a la relación entre los intervalos y la clase.

Global versus Local

- Los métodos **globales** usan todo el espacio de instancias para el proceso de discretización.
- Los métodos **locales** usan solo un subconjunto de las instancias para el proceso de discretización. Se relacionan con la discretización dinámica. Un atributo cualquiera puede ser discretizado en distintos intervalos (arboles).
- Las técnicas **globales** son más eficientes, porque solamente se usa una discretización a través de todo el proceso de data mining, pero las técnicas locales podrían provocar el descubrimiento de puntos de corte más útiles.

Clasificacion de los metodos de discretization



Evaluación de los métodos de discretización

- El número total de intervalos generados. Un número pequeño de intervalos generados es bueno hasta cierto punto.
- El número de inconsistencias en el conjunto de datos discretizado. La inconsistencia debe disminuir.
- La precision de prediccion. El proceso de discretización no debe tener un gran efecto en el tasa de error de mala clasificación.

Programas para Discretizacion

En R se puede usar funciones de la libreria dprep o la libreria Discretization (2010), aunque esta ultima solo contiene dos de los metodos discutidos en este curso: discretizacion por entropia, discretizacion ChiMerge.
En Python, el modulo Pandas tiene algunos metodos de binning. Otros metodos estan en el paquete Orange.
Rapidminer hace discretizacion uando intervalos iguales (binning) y intervalos con igual frecuencia y discretizacion por entropia.
Se elige el proceso Cleansing y de alli se elige bining y luego el metodo de Discretization deseado.

WEKA hace discretizacion uando intervalos iguales (binning) y intervalos con igual frecuencia, para esto se sigue la secuencia filter>attribute>unsupervised>discretize
Weka tambien hace discretizacion por entropia, para esto se sigue la secuencia: filter>attribute>supervised>discretize.

Intervalos de igual amplitud (binning)

- Dividir el rango de cada variable en k intervalos de igual tamaño.
- si *A* es el menor valor y *B* el mayor valor del atributo, el ancho de los intervalos será:

$$W = (B-A) / k$$

- Los límites de los intervalos ademas de A y B seran:

$$A+W, A+2W, \ldots, A + (k-1)W$$

- Formas de determinar k:
  - Fórmula de Sturges: $k=\log_2(n+1)$, n: número de observaciones.
  - Fórmula de Friedman-Diaconis: $W=2*IQR*n^{-1/3}$, donde IQR = Q3 - Q1. Luego k=(B-A)/W
  - Fórmula de Scott: $W=3.5*s*n^{-1/3}$, donde s es la desviación estándar. Luego k=(B-A)/W.

  Este metodo es considerado como no supervisado, global y estatico.
- Problemas
  - (a) No supervisado
  - (b) De donde proviene k?
  - (c) Sensitivo a outliers.

Ejemplo: Discretización usando intervalos de igual amplitud (libreria Dprep)

```
> dbupa=disc.ew(bupa,1:6,out="num")
> table(dbupa[,1])
 1  7  8  9 10 11 12 13 14 15 16 17 18
 1  3  2 22 39 52 60 79 35 23 19  7  3
> table(dbupa[,2])
 1  2  3  4  5  6  7  8  9 10 11 12 13
 1  6 30 62 80 58 43 27 21  8  4  3  2
> table(dbupa[,3])
  1   2   3   4   5   6   7   8   9  11  12  16
 24 105 114  48  19  14   8   3   5   1   1   3
> table(dbupa[,4])
  1   2   3   4   5   6   7   8   9  10  11  13  14  15  16
  3  21  76 108  71  23  16  10   6   3   3   1   1   1   2
> table(dbupa[,5])
  1   2   3   4   5   6   7   8   9  11  12  15
172  83  39  17   9  11   3   3   1   5   1   1
> table(dbupa[,6])
  1   2   3   4   5   6   7   8  10  11  13
134  56  50  57   6  23  10   4   1   2   2
```

Intervalos de igual frecuencia

- Dividir el rango en *k* intervalos
- Cada intervalo contendrá aproximadamente el mismo número de instancias.
- El proceso de discretización ignora la información de la clase.

Ejemplo: Intervalos de igual frecuencia (dprep)

```
>args(disc.ef)
function (data, varcon, k, out = c("symb",
"num")) NULL
>dbupa=disc.ef(bupa,1:6,10,out="num")
> table(dbupa[,1])
 1  2  3  4  5  6  7  8  9 10
35 35 35 35 35 35 35 35 35 30
> table(dbupa[,2])
1  2  3  4  5  6  7  8  9 10
35 35 35 35 35 35 35 35 35 30
> table(dbupa[,3])
1  2  3  4  5  6  7  8  9 10
35 35 35 35 35 35 35 35 35 30
> table(dbupa[,4])
 1  2  3  4  5  6  7  8  9 10
35 35 35 35 35 35 35 35 35 30
```

Método 1R

- Desarrollado por Holte (1993)
- Es un método de discretización supervisado que usa intervalos de valores (binning).
- Para cada atributo, después de ordenar sus valores, el rango de valores continuos es dividido en intervalos disjuntos y los límites de esos intervalos son ajustados en base a las etiquetas de la clase asociada con los valores del atributo.
- El ajuste de los límites continua hasta que el valor que sigue corresponda a una clase distinta a la clase mayoritaria en el intervalo adyacente.
- Cada intervalo debe contener un número mínimo dado de instancias (15 por defecto) con excepción del último.

Ejemplo de 1R

Datos ordenados

bupat[1:50,1] #los 50 primeros valores de la primera variable

[1] 65 78 79 79 81 81 82 82 82 82 82 82 82 83 83 83 83 83 83 84 84 84 84 84 84

[26] 84 84 84 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 86 86 86 86 86

#Asignando los datos a las clases y determinando la clase mayoritaria.Cada clase tiene tamano minimo de 6

- bupat[1:50,2]

Ejemplo de 1R

[1] 2 1 2 2 2 1* 1 2 1 2 2 2 2 2 *1 2 2 2 1 2 2 *1 1 2 2 1 2 1 *2 2 2 2 2 2 2 2
2*

2 2 2 1 2

[39] 1 1 2 2 2 2 2 2 *1 1 2 1

2 1

#Juntando los intervalos adyacentes que tienen la misma clase mayoritaria.

#Datos discretizados

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 4 4 4 4

Discretizacion 1R (libreria dprep)

```
> args(disc.1r)
function (data, convar, binsize = 15, out = c("symb", "num"))
NULL
>dbupa=disc.1r(bupa,1:6,binsize=10,out="num")
>table(dbupa[,1])
1 2 3 4 5
90 29 25 35 166
>table(dbupa[,2])
1 2 3 4 5 6 7 8 9 10
10 105 16 29 19 14 14 30 29 79
>table(dbupa[,3])
 1 2 3 4 5 6 7 8 9 10
87 17 14 22 44 36 49 10 12 54
>table(dbupa[,4])
 1 2 3 4
37 25 66 217
```

Discretización basada en Entropía

- Fayyad and Irani (1993)
- Los métodos basados en entropía utilizan la información existente de la clase en los datos.
- La entropía (o contenido de información) es calculada en base a la clase. Intuitivamente, encuentra la mejor partición de cada atributo de tal forma que las divisiones sean las mas puras posible, i.e. la mayoría de los valores en una division corresponden a la misma clase. Formalmente, es caracterizado por encontrar la partición con la máxima ganancia de información.
- Es un metodo de discretizacion supervisado, global y estatico.

Discretización basada en Entropía (cont)

- Sea S el siguiente conjunto de 9 pares (atributo-valor, clase), S = {(0,Y), (4,Y), (12,Y), (16,N), (16,N), (18,Y), (24,N), (26,N), (28,N)}. Sea p_1 = 4/9 la fracción de pares con clase=Y, y p_2 = 5/9 la fracción de pares con clase=N.
- La entropía (o contenido de información) para S se define como:

  Entropy(S) = - p_1*$\log_2(p_1)$ – p_2*$\log_2(p_2)$.

En este caso, Entropy(S)=0.991076.

- Si la entropía es pequeña, entonces el conjunto es relativamente puro. El valor más pequeño posible es 0.
- Si la entropía es grande, entonces el conjunto esta muy mezclado. El valor más grande posible de entropia es 1, el cual es obtenido cuando p_1=p_2=0.5

Discretización basada en Entropía (cont)

- Si el conjunto de muestras S es particionado en dos intervalos S_1 y S_2 usando el punto de corte T, la entropía después de particionar es:

$$E(S,T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

donde | | denota cardinalidad. El punto de corte T se escoge de los puntos medios de los valores del atributo, i e: {2, 8, 14, 16, 17, 21, 25, 27}. Por ejemplo si T: valor de atributo=14

S_1= (0,Y), (4,Y), (12,Y) y S_2= (16,N), (16,N), (18,Y), (24,N), (26,N), (28,N)

E(S,T)=(3/9)*E(S_1)+(6/9)*E(S_2)=3/9*0+(6/9)* 0.6500224

E(S,T)=.4333

Ganancia de información de la partición, Gain(S,T) = Entropy(S) - E(S,T).

Gain=.9910-.4333=.5577

Discretización basada en Entropía (cont)

Igualmente, para T: v=21 se obtiene

Ganancia de Información=.9910-.6121=.2789.

Por lo que v=14 es una mejor partición.

- El objetivo de este algoritmo es encontrar la partición con la máxima ganancia de información. La ganancia máxima se obtiene cuando E(S,T) es mínima.
- La mejor partición (es) se encuentran examinando todas las posibles particiones y seleccionando la óptima. El punto de corte que minimiza la función de entropía sobre todos los posibles puntos de corte se selecciona como una discretización binaria.
- El proceso es aplicado recursivamente a particiones obtenidas hasta que se cumpla algún criterio de parada, e.g.,

$$Ent(S) - E(T,S) > \delta$$

Discretización basada en Entropía (cont)

Donde:

$$\partial > \frac{\log(N-1)}{N} + \frac{\Delta(T,S)}{N}$$

y,

$$\Delta(S,T) = \log_2(3^c - 2) - [cEnt(S) - c_1 Ent(S_1) - c_2 Ent(S_2)]$$

Aquí c es el número de clases en S, c_1 es el número de clases en S_1 y c_2 es el número de clases en S_2. Esto es llamado el Principio de Longitud de Descripción Mínima (MDLP)

Discretización usando Entropía con MDL(libreria dprep)

```
>args(disc.mentr)
function (data, varcon, out = c("symb", "num"))
NULL
>dbupa=disc.mentr(bupa,1:6,out="num")
The number of partitions for var 1 is : 1
The cut points are: [1] 0
The number of partitions for var 2 is : 1
The cut points are: [1] 0
The number of partitions for var 3 is : 1
The cut points are: [1] 0
The number of partitions for var 4 is : 1
The cut points are: [1] 0
The number of partitions for var 5 is : 2
The cut points are: [1] 20.5
The number of partitions for var 6 is : 1
The cut points are: [1] 0
```

ChiMerge (Kerber92)

- Este método de discretización utiliza un enfoque de juntar intervalos.
- Caracteristicas de ChiMerge:
  - Las frecuencias relativas de clase deben ser bastante parecidas dentro de un intervalo (de lo contrario se debe dividir el intervalo)
  - Dos intervalos adyacentes no deben tener similares frecuencias relativas de clase (de lo contrario se debe juntar)

Prueba χ^2 y Discretización

- χ^2 es una medida estadística para probar la hipótesis de que dos atributos discretos son estadísticamente independientes.
- Para dos intervalos adyacentes, si la prueba χ^2 concluye que la clase es independiente de los intervalos, los intervalos se deben juntar. Si la prueba χ^2 concluye que no son independientes, i.e., la diferencia en frecuencia relativa de la clase es estadísticamente significativa, los dos intervalos deben continuar separados.

Tabla de contingencia

	Clase 1	Clase 2	Total
Intervalo I	A_{11}	A_{12}	R_1
Intervalo II	A_{21}	A_{22}	R_2
Total	C_1	C_2	N

Calculando χ^2

- Este valor puede ser obtenido así:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

k = número de clases

A_{ij} = número de datos en el i-ésimo intervalo, j-ésima clase

E_{ij} = frecuencia esperada de A_{ij}

$= (R_i * C_j) / N$

R_i = número de datos en el i-ésimo intervalo

C_j = número de datos en la j-ésima clase

N = número total de datos en los dos intervalos

Si Eij=0 entonces asignar a Eij un valor pequeño, por ejemplo 0.1

ChiMerge – El algoritmo

- 1)Obtener el valor de χ^2 para cada par de intervalos adyacentes.
- 2) Juntar el par de intervalos adyacentes que tengan el menor valor de χ^2
- Repetir pasos 1 y 2 hasta que los valores χ^2 de todos los pares adyacentes excedan un valor dado (threshold)
- Threshold: es determinado por el *nivel de significancia* y el *grado de libertad* = número de clases - 1

Ejemplo

Sample:	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Ejemplo (cont.)

Los valores de particionamiento iniciales son los puntos medios de la variable F
El minimo valor de χ2 esta en [7.5,8.5] y [8.5,10], con clase K=1

	Clase 1	Clase 2	Sumas
Intervalo [7.5,8.5]	1	0	1
Intervalo [8.5,10]	1	0	1
Sumas	2	0	2

Asi, E11=1, E12=0~0.1, E21=1, E22=0~0.1, df=grados de libertad=1
Umbral (para α=10%)=2.706
X^2=0.2. las diferencias no son significantes →junta

Ejemplo (cont.)

Tablas de contingencia para los intervalos [0,10] y [10,42]

	Clase 1	Clase 2	Sumas
Intervalo [0,10]	4	1	5
Intervalo [10,42]	1	3	4
Sumas	5	4	9

Asi E11=2.78, E12=2.22, E21=2.22 E22=1.78, df=grados de libertad=1.
Umbral (para α=10%)=2.706
X^2=2.72. Las diferencias son significantes → No juntar

Ejemplo (cont.)

El resultado final son tres intervalos [0,10],[10,42],[42,60]

```
chiMerge(mat,1,.1)
      f k
 [1,] 1 1
 [2,] 1 2
 [3,] 1 1
 [4,] 1 1
 [5,] 1 1
 [6,] 2 2
 [7,] 2 2
 [8,] 2 1
 [9,] 2 2
[10,] 3 1
[11,] 3 1
[12,] 3 1
>
```

Ejemplo: Discretización de Bupa

```
args(chiMerge)
function (data, varcon, alpha = 0.1, out = c("symb", "num"))
NULL
> dbupa=chiMerge(bupa,1:6,.05,out="num")
> table(dbupa[,1])
  1   2   3
 90 250   5
> table(dbupa[,2])
  1   2   3   4   5   6   7   8   9  10  11  12
  3   4   3  42   9  46 100  30   7   6  16  79
> table(dbupa[,3])
  1   2   3   4   5
 24  21 284   7   9
> table(dbupa[,4])
  1   2   3   4   5   6   7   8
208  20  58   9  35   9   1   5
> table(dbupa[,5])
  1   2   3   4   5   6   7   8   9
  9  69  11  14  37 113  34   3  55
> table(dbupa[,6])
  1   2   3   4
190  67  83   5
```

Efectos de la Discretización

- Los resultados experimentales indican que después de la discretización
  - El tamaño del conjunto de datos puede ser reducido (Rough sets).
  - La exactitud de la clasificación puede mejorar