Package ‘discretization’

February 19, 2015

Type Package

Title Data preprocessing, discretization for classification.

Version 1.0-1

Date 2010-12-02

Author HyunJi Kim

Maintainer HyunJi Kim <polaris7867@gmail.com>

Description This package is a collection of supervised discretization algorithms. It can also be grouped in terms of top-down or bottom-up, implementing the discretization algorithms.

License GPL

LazyLoad yes

Repository CRAN

Date/Publication 2012-10-29 08:58:35

NeedsCompilation no

R topics documented:

---

discretization-package   *Data preprocessing, discretization for classification.*

Description

This package is a collection of supervised discretization algorithms. It can also be grouped in terms of top-down or bottom-up, implementing the discretization algorithms.

Details

Package:	discretization
Type:	Package
Version:	1.0-1
Date:	2010-12-02
License: GPL LazyLoad:	yes

Author(s)

Maintainer: HyunJi Kim <polaris7867@gmail.com>

References

Choi, B. S., Kim, H. J., Cha, W. O. (2011). A Comparative Study on Discretization Algorithms for Data Mining, Communications of the Korean Statistical Society, to be published.

Chmielewski, M. R. and Grzymala-Busse, J. W. (1996). Global Discretization of Continuous Attributes as Preprocessing for Machine Learning, *International journal of approximate reasoning*, **Vol. 15, No. 4**, 319–331.

Fayyad, U. M. and Irani, K. B.(1993). Multi-interval discretization of continuous-valued attributes for classification learning, *Artificial intelligence*, **13**, 1022–1027.

Gonzalez-Abril, L., Cuberos, F. J., Velasco, F. and Ortega, J. A. (2009), Ameva: An autonomous discretization algorithm,*Expert Systems with Applications*, **36**, 5327–5332.

Kerber, R. (1992). ChiMerge : Discretization of numeric attributes, *In Proceedings of the Tenth National Conference on Artificial Intelligence*, 123–128.

Kurgan, L. A. and Cios, K. J. (2004). CAIM Discretization Algorithm, *IEEE Transactions on knowledge and data engineering*, **16**, 145-153.

Liu, H. and Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes, *Tools with Artificial Intelligence*, 388–391.

Liu, H. and Setiono, R. (1997). Feature selection and discretization, *IEEE transactions on knowledge and data engineering*, **9**, 642–645.

Pawlak, Z. (1982). Rough Sets, *International Journal of Computer and Information Sciences*, **vol.11, No.5**, 341–356.

Su, C. T. and Hsu, J. H. (2005). An Extended Chi2 Algorithm for Discretization of Real Value Attributes, *IEEE transactions on knowledge and data engineering*, **17**, 437–441.

Tay, F. E. H. and Shen, L. (2002). Modified Chi2 Algorithm for Discretization, *IEEE Transactions on knowledge and data engineering*, **14**, 666–670.

Tsai, C. J., Lee, C. I. and Yang, W. P. (2008). A discretization algorithm based on Class-Attribute Contingency Coefficient, *Information Sciences*, **178**, 714–731.

Ziarko, W. (1993). Variable Precision Rough Set Model, *Journal of computer and system sciences*, **Vol. 46, No. 1**, 39–59.

`ameva` *Auxiliary function for Ameva algorithm*

Description

This function is required to compute the ameva value for Ameva algorithm.

Usage

```
ameva(tb)
```

Arguments

`tb`	a vector of observed frequencies, $k * l$

Details

This function implements the Ameva criterion proposed in Gonzalez-Abril, Cuberos, Velasco and Ortega (2009) for Discretization. An autonomous discretization algorithm(Ameva) implements in `disc.Topdown(data,method=1)` It uses a measure based on chi^2 as the criterion for the optimal discretization which has the minimum number of discrete intervals and minimum loss of class variable interdependence. The algorithm finds local maximum values of Ameva criterion and a stopping criterion.

Ameva coefficient is defined as follows:

$$Ameva(k) = \frac{\chi^2(k)}{k * (l-1)}$$

for $k, l >= 2$, *k* is a number of intervals, *l* is a number of classes.

This value calculates in contingency table between class variable and discrete interval, row matrix representing the class variable and each column of discrete interval.

Value

`val`	numeric value of Ameva coefficient

Author(s)

HyunJi Kim `<polaris7867@gmail.com>`

References

Gonzalez-Abril, L., Cuberos, F. J., Velasco, F. and Ortega, J. A. (2009) Ameva: An autonomous discretization algorithm, *Expert Systems with Applications*, **36**, 5327–5332.

See Also

`disc.Topdown`, `topdown`, `insert`, `findBest` and `chiSq`.

Examples

```
#--Ameva criterion value
a=c(2,5,1,1,3,3)
m=matrix(a,ncol=3,byrow=TRUE)
ameva(m)
```

---

cacc — *Auxiliary function for CACC discretization algorithm*

Description

This function is requied to compute the cacc value for CACC discretization algorithm.

Usage

```
cacc(tb)
```

Arguments

`tb`	a vector of observed frequencies

Details

The Class-Attribute Contingency Coefficient(CACC) discretization algorithm implements in `disc.Topdown(data,method=2`

The cacc value is defined as

$$cacc = \sqrt{\frac{y}{y+M}}$$

for

$$y = \chi^2/log(n)$$

M is the total number of samples, n is a number of discretized intervals. This value calculates in contingency table between class variable and discrete interval, row matrix representing the class variable and each column of discrete interval.

Value

`val`	numeric of cacc value

Author(s)

HyunJi Kim `<polaris7867@gmail.com>`

References

Tsai, C. J., Lee, C. I. and Yang, W. P. (2008). A discretization algorithm based on Class-Attribute Contingency Coefficient, *Information Sciences*, **178**, 714–731.

See Also

`disc.Topdown`, `topdown`, `insert`, `findBest` and `chiSq`.

Examples

```
#----Calculating cacc value (Tsai, Lee, and Yang (2008))
a=c(3,0,3,0,6,0,0,3,0)
m=matrix(a,ncol=3,byrow=TRUE)
cacc(m)
```

`caim` *Auxiliary function for caim discretization algorithm*

Description

This function is required to compute the CAIM value for CAIM iscretization algorithm.

Usage

```
caim(tb)
```

Arguments

tb a vector of observed frequencies

Details

The Class-Attrivute Interdependence Maximization(CAIM) discretization algorithm implements in `disc.Topdwon(data,method=1)`. The CAIM criterion measures the dependency between the class variable and the discretization variable for attribute, and is defined as :

$$CAIM = \frac{\sum_{r=1}^{n} \frac{max_r^2}{M_+r}}{n}$$

for $r = 1, 2, ..., n$, max_r is the maximum value within the rth column of the quanta matrix. M_{+r} is the total number of continuous values of attribute that are within the interval(Kurgan and Cios (2004)).

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Kurgan, L. A. and Cios, K. J. (2004). CAIM Discretization Algorithm, *IEEE Transactions on knowledge and data engineering*, **16**, 145–153.

See Also

disc.Topdown, topdown, insert, findBest.

Examples

```
#----Calculating caim value
a=c(3,0,3,0,6,0,0,3,0)
m=matrix(a,ncol=3,byrow=TRUE)
caim(m)
```

---

chi2 *Discretization using the Chi2 algorithm*

Description

This function performs Chi2 discretization algorithm. Chi2 algorithm automatically determines a proper Chi-sqaure(χ^2) threshold that keeps the fidelity of the original numeric dataset.

Usage

```
chi2(data, alp = 0.5, del = 0.05)
```

Arguments

`data`	the dataset to be discretize
`alp`	significance level; α
`del`	$Inconsistency(data) < \delta$, (Liu and Setiono(1995))

Details

The Chi2 algorithm is based on the χ^2 statistic, and consists of two phases. In the first phase, it begins with a high significance level(sigLevel), for all numeric attributes for discretization. Each attribute is sorted according to its values. Then the following is performed: **phase 1.** calculate the χ^2 value for every pair of adjacent intervals (at the beginning, each pattern is put into its own interval that contains only one value of an attribute); **pahse 2.** merge the pair of adjacent intervals with the lowest χ^2 value. Merging continues until all pairs of intervals have χ^2 values exceeding the parameter determined by sigLevel. The above process is repeated with a decreased sigLevel until an *inconsistency rate*(δ), `incon()`, is exceeded in the discretized data(Liu and Setiono (1995)).

Value

`cutp`	list of cut-points for each variable
`Disc.data`	discretized data matrix

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Liu, H. and Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes, *Tools with Artificial Intelligence*, 388–391.

Liu, H. and Setiono, R. (1997). Feature selection and discretization, *IEEE transactions on knowledge and data engineering*, **Vol.9, no.4**, 642–645.

See Also

`value`, `incon` and `chiM`.

Examples

```
data(iris)
#---cut-points
chi2(iris,0.5,0.05)$cutp

#--discretized dataset using Chi2 algorithm
chi2(iris,0.5,0.05)$Disc.data
```

chiM *Discretization using ChiMerge algorithm*

Description

This function implements ChiMerge discretization algorithm.

Usage

```
chiM(data, alpha = 0.05)
```

Arguments

`data`	numeric data matrix to discretized dataset
`alpha`	significance level; α

Details

The ChiMerge algorithm follows the axis of bottom-up. It uses the χ^2 statistic to determine if the relative class frequencies of adjacent intervlas are distinctly different or if they are similar enough to justify merging them into a single interval(Kerber, R. (1992)).

Value

`cutp`	list of cut-points for each variable
`Disc.data`	discretized data matrix

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Kerber, R. (1992). ChiMerge : Discretization of numeric attributes, *In Proceedings of the Tenth National Conference on Artificial Intelligence*, 123–128.

See Also

`chiSq`, `value`.

Examples

```
#--Discretization using the ChiMerge method
data(iris)
disc=chiM(iris,alpha=0.05)

#--cut-points
disc$cutp
```

```
#--discretized data matrix
disc$Disc.data
```

---

chiSq *Auxiliary function for discretization using Chi-square statistic*

Description

This function is required to perform the discretization based on Chi-square statistic(CACC, Ameva, ChiMerge, Chi2, Modified Chi2, Extended Chi2).

Usage

```
chiSq(tb)
```

Arguments

tb	a vector of observed frequencies

Details

The formula for computing the χ^2 value is

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

$k =$ number of (no.) classes, $A_{ij} =$ no. patterns in the ith interval, jth class, $R_i =$ no. patterns in the jth class = $\sum_{j=1}^{k} A_{ij}$, $C_j =$ no. patterns in the $j$the class = $\sum_{i=1}^{2} A_{ij}$, $N =$ total no. patterns = $\sum_{i=1}^{2} R_i j$, $E_{ij} =$ expected frequency of $A_{ij} = R_i * C_j / N$. If either R_i or C_j is 0, E_{ij} is set to 0.1. The degree of freedom of the χ^2 statistic is on less the number of classes.

Value

val	χ^2 value

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Kerber, R. (1992). ChiMerge : Discretization of numeric attributes, *In Proceedings of the Tenth National Conference on Artificial Intelligence*, 123–128.

See Also

cacc, ameva, chiM, chi2, modChi2 and extendChi2.

Examples

```
#----Calulate Chi-Square
b=c(2,4,1,2,5,3)
m=matrix(b,ncol=3)
chiSq(m)
chisq.test(m)$statistic
```

---

cutIndex *Auxiliary function for the MDLP*

Description

This function is required to perform the Minimum Description Length Principle.`mdlp`

Usage

```
cutIndex(x, y)
```

Arguments

`x`	a vector of numeric value
`y`	class variable vector

Details

This function computes the best cut index using entropy

Author(s)

HyunJi Kim <polaris7867@gmail.com>

See Also

`cutPoints`, `ent`, `mergeCols`, `mdlStop`, `mylog`, `mdlp` .

cutPoints — *Auxiliary function for the MDLP*

Description

This function is required to perform the Minimum Description Length Principle.mdlp

Usage

```
cutPoints(x, y)
```

Arguments

x	a vector of numeric value
y	class variable vector

Author(s)

HyunJi Kim <polaris7867@gmail.com>

See Also

cutIndex, ent, mergeCols, mdlStop, mylog, mdlp .

disc.Topdown — *Top-down discretization*

Description

This function implements three top-down discretization algorithms(CAIM, CACC, Ameva).

Usage

```
disc.Topdown(data, method = 1)
```

Arguments

data	numeric data matrix to discretized dataset
method	1: CAIM algorithm, 2: CACC algorithm, 3: Ameva algorithm.

Value

cutp	list of cut-points for each variable(minimun value, cut-points and maximum value)
Disc.data	discretized data matrix

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Gonzalez-Abril, L., Cuberos, F. J., Velasco, F. and Ortega, J. A. (2009) Ameva: An autonomous discretization algorithm, *Expert Systems with Applications*, **36**, 5327–5332.

Kurgan, L. A. and Cios, K. J. (2004). CAIM Discretization Algorithm, *IEEE Transactions on knowledge and data engineering*, **16**, 145–153.

Tsai, C. J., Lee, C. I. and Yang, W. P. (2008). A discretization algorithm based on Class-Attribute Contingency Coefficient, *Information Sciences*, **178**, 714–731.

See Also

topdown, insert, findBest, findInterval, caim, cacc, ameva

Examples

```
##---- CAIM discretization ----
##----cut-potins
cm=disc.Topdown(iris, method=1)
cm$cutp
##----discretized data matrix
cm$Disc.data

##---- CACC discretization----
disc.Topdown(iris, method=2)

##---- Ameva discretization ----
disc.Topdown(iris, method=3)
```

---

ent *Auxiliary function for the MDLP*

Description

This function is required to perform the Minimum Description Length Principle.mdlp

Usage

```
ent(y)
```

Arguments

y	class variable vector

Author(s)

HyunJi Kim <polaris7867@gmail.com>

See Also

cutPoints, ent, mergeCols, mdlStop, mylog, mdlp .

---

extendChi2 *Discretization of Numeric Attributes using the Extended Chi2 algorithm*

Description

This function implements Extended Chi2 discretization algorithm.

Usage

```
extendChi2(data, alp = 0.5)
```

Arguments

data	data matrix to discretized dataset
alp	significance level; α

Details

In the extended Chi2 algorithm, inconsistency checking($InConCheck(data) < \delta$) of the Chi2 algorithm is replaced by the lease upper bound ξ(Xi()) after each step of discretization ($\xi_{discretized} < \xi_{original}$). It uses as the stopping criterion.

Value

cutp	list of cut-points for each variable
Disc.data	discretized data matrix

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Su, C. T. and Hsu, J. H. (2005). An Extended Chi2 Algorithm for Discretization of Real Value Attributes, *IEEE transactions on knowledge and data engineering*, **17**, 437–441.

See Also

chiM, Xi

Examples

```
data(iris)
ext=extendChi2(iris,0.5)
ext$cutp
ext$Disc.data
```

`findBest` *Auxiliary function for top-down discretization*

Description

This function is required to perform the `disc.Topdown()`.

Usage

```
findBest(x, y, bd, di, method)
```

Arguments

`x`	a vector of numeric value
`y`	class variable vector
`bd`	current cut points
`di`	candidate cut-points
`method`	each `method` number indicates three top-down discretization. `1` for CAIM algorithm, `2` for CACC algorithm, `3` for Ameva algorithm.

Author(s)

HyunJi Kim <polaris7867@gmail.com>

See Also

`topdown`, `insert` and `disc.Topdown`.

incon — *Computing the inconsistency rate for Chi2 discretization algorithm*

Description

This function computes the inconsistency rate of dataset.

Usage

```
incon(data)
```

Arguments

`data`	dataset matrix

Details

The inconsistency rate of dataset is calculated as follows: (1) two instances are considered inconsistent if they match except for their class labels; (2) for all the matching instances (without considering their class labels), the inconsistency count is the number of the instances minus the largest number of instnces of class labels; (3) the inconsistency rate is the sum of all the inconsistency counts divided by the total number of instances.

Value

`inConRate`	the inconsistency rate of the dataset

Author(s)

HyunJi Kim `<polaris7867@gmail.com>`

References

Liu, H. and Setiono, R. (1995), Chi2: Feature selection and discretization of numeric attributes , *Tools with Artificial Intelligence*, 388–391.

Liu, H. and Setiono, R. (1997), Feature selection and discretization, *IEEE transactions on knowledge and data engineering*, **Vol.9, no.4**, 642–645.

See Also

`chi2`

Examples

```
##---- Calculating Inconsistency ----
data(iris)
disiris=chiM(iris,alpha=0.05)$Disc.data
incon(disiris)
```

insert *Auxiliary function for Top-down discretization*

Description

This function is required to perform the `disc.Topdown()`.

Usage

```
insert(x, a)
```

Arguments

`x`	cut-point
`a`	a vector of minimum, maximum value

Author(s)

HyunJi Kim <polaris7867@gmail.com>

See Also

`topdown`, `findBest` and `disc.Topdown` .

LevCon *Auxiliary function for the Modified Chi2 discretization algorithm*

Description

This function computes the level of consistency, is required to perform the Modified Chi2 discretization algorithm.

Usage

```
LevCon(data)
```

Arguments

`data`	discretized data matrix

Value

`LevelConsis`	Level of Consistency value

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Tay, F. E. H. and Shen, L. (2002). Modified Chi2 Algorithm for Discretization, *IEEE Transactions on knowledge and data engineering*, **Vol. 14, No. 3**, 666–670.

Pawlak, Z. (1982). Rough Sets, *International Journal of Computer and Information Sciences*, **vol.11, No.5**, 341–356.

Chmielewski, M. R. and Grzymala-Busse, J. W. (1996). Global Discretization of Continuous Attributes as Preprocessing for Machine Learning, *International journal of approximate reasoning*, **Vol. 15, No. 4**, 319–331.

See Also

`modChi2`

`mdlp` *Discretization using the Minimum Description Length Principle(MDLP)*

Description

This function discretizes the continuous attributes of data matrix using entropy criterion with the Minimum Description Length as stopping rule.

Usage

```
mdlp(data)
```

Arguments

`data`	data matrix to be discretized dataset

Details

Minimum Discription Length Principle

Value

`cutp`	list of cut-points for each variable
`Disc.data`	discretized data matrix

Author(s)

HyunJi Kim `polaris7867@gmail.com`

References

Fayyad, U. M. and Irani, K. B.(1993). Multi-interval discretization of continuous-valued attributes for classification learning, *Artificial intelligence*, **13**, 1022–1027.

See Also

cutIndex, cutPoints, ent, mergeCols, mdlStop, mylog .

Examples

```
data(iris)
mdlp(iris)$Disc.data
```

---

mdlStop *Auxiliary function for performing discretization using MDLP*

Description

This function determines cut criterion based on Fayyad and Irani Criterion, is required to perform the minimum description length principle.

Usage

```
mdlStop(ci, y, entropy)
```

Arguments

ci	cut index
y	class variable
entropy	this value is calculated by `cutIndex()`

Details

Minimum description Length Principle Criterion

Value

gain	numeric value

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Fayyad, U. M. and Irani, K. B.(1993). Multi-interval discretization of continuous-valued attributes for classification learning, *Artificial intelligence*, **13**, 1022–1027.

See Also

cutPoints, ent, mergeCols, cutIndex, mylog, mdlp .

`mergeCols` *Auxiliary function for performing discretization using MDLP*

Description

This function merges the columns having observation numbers equal to 0, required to perform the minimum discription length principle.

Usage

```
mergeCols(n, minimum = 2)
```

Arguments

`n`	table, column: intervals, row: variables
`minimum`	min # observations in col or row to merge

Author(s)

HyunJi Kim <polaris7867@gmail.com>

See Also

`cutPoints`, `ent`, `cutIndex`, `mdlStop`, `mylog`, `mdlp` .

`modChi2` *Discretization of Nemeric Attributes using the Modified Chi2 method*

Description

This function implements the Modified Chi2 discretization algorithm.

Usage

```
modChi2(data, alp = 0.5)
```

Arguments

`data`	numeric data matrix to discretized dataset
`alp`	significance level, α

Details

In the modified Chi2 algorithm, inconsistency checking($InConCheck(data) < \delta$) of the Chi2 algorithm is replaced by maintaining the level of consistency L_c after each step of discretization ($L_{c-discretized} < L_{c-original}$). this inconsistency rate as the stopping criterion.

Value

`cutp`	list of cut-points for each variable
`Disc.data`	discretized data matrix

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Tay, F. E. H. and Shen, L. (2002). Modified Chi2 Algorithm for Discretization, *IEEE Transactions on knowledge and data engineering*, **14**, 666–670.

See Also

`LevCon`

Examples

```
data(iris)
modChi2(iris, alp=0.5)$Disc.data
```

`mylog` *Auxiliary function for performing discretization using MDLP*

Description

This function is required to perform the minimum discription length principle, `mdlp()`.

Usage

```
mylog(x)
```

Arguments

`x`	a vector of numeric value

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Fayyad, U. M. and Irani, K. B.(1993). Multi-interval discretization of continuous-valued attributes for classification learning, *Artificial intelligence*, **Vol. 13**, 1022–1027.

See Also

`mergeCols`, `ent`, `cutIndex`, `cutPoints`, `mdlStop` and `mdlp`.

---

topdown — *Auxiliary function for performing top-down discretization algorithm*

Description

This function is required to perform the `disc.Topdown()`.

Usage

```
topdown(data, method = 1)
```

Arguments

`data`	numeric data matrix to discretized dataset
`method`	`1`: CAIM algorithm, `2`: CACC algorithm, `3`: Ameva algorithm.

Author(s)

HyunJi Kim <`polaris7867@gmail.com`>

References

Gonzalez-Abril, L., Cuberos, F. J., Velasco, F. and Ortega, J. A. (2009) Ameva: An autonomous discretization algorithm, *Expert Systems with Applications*, **36**, 5327–5332.

Kurgan, L. A. and Cios, K. J. (2004). CAIM Discretization Algorithm, *IEEE Transactions on knowledge and data engineering*, **16**, 145–153.

Tsai, C. J., Lee, C. I. and Yang, W. P. (2008). A discretization algorithm based on Class-Attribute Contingency Coefficient, *Information Sciences*, **178**, 714–731.

See Also

`insert`, `findBest` and `disc.Topdown` .

---

value — *Auxiliary function for performing the ChiMerge discretization*

Description

This function is called by ChiMerge diacretization fucntion, `chiM()`.

Usage

```
value(i, data, alpha)
```

Arguments

`i`	ith variable in data matrix to discretized
`data`	numeric data matrix
`alpha`	significance level; α

Value

`cuts`	list of cut-points for any variable
`disc`	discretized ith variable and data matrix of other variables

Author(s)

HyunJi Kim `<polaris7867@gmail.com>`

References

Kerber, R. (1992). ChiMerge : Discretization of numeric attributes, *In Proceedings of the Tenth National Conference on Artificial Intelligence*, 123–128.

See Also

`chiM`.

Examples

```
data(iris)
value(1,iris,0.05)
```

`Xi` *Auxiliary function for performing the Extended Chi2 discretization algorithm*

Description

This function is the ξ, required to perform the Extended Chi2 discretization algorithm.

Usage

```
Xi(data)
```

Arguments

`data`	data matrix

Details

The following equality is used for calculating the least upper bound(ξ) of the data set(Chao and Jyh-Hwa (2005)).

$$\xi(C, D) = max(m_1, m_2)$$

where C is the equivalence relation set, D is the decision set, and $C^* = \{E_1, E_2, \dots, E_n\}$ is the equivalence classes. $m_1 = 1 - min\{c(E, D)|E \in C^*$ and $0.5 < c(E, D)\}$, $m_2 = 1 - max\{c(E, D)|E \in C^*$ and $c(E, D) < 0.5\}$.

$$c(E, D) = 1 - \frac{card(E \cap D)}{card(E)}$$

$card$ denotes set cardinality.

Value

`Xi`	numeric value, ξ

Author(s)

HyunJi Kim <polaris7867@gmail.com>

References

Chao-Ton, S. and Jyh-Hwa, H. (2005). An Extended Chi2 Algorithm for Discretization of Real Value Attributes, *IEEE transactions on knowledge and data engineering*, **Vol. 17, No. 3**, 437–441.

Ziarko, W. (1993). Variable Precision Rough Set Model, *Journal of computer and system sciences*, **Vol. 46, No. 1**, 39–59.

See Also

`extendChi2`

Index