Association Rules: nagdmc_assoc

Purpose

nagdmc_assoc computes association rules for items sorted in ascending order.

Declaration

```
#include <nagdmc.h>

void nagdmc_assoc(long rec1, long nrec, long dblk, long ni, long nit[],
                  long data[], int supptype, double minsupp, double minconf,
                  long maxnir, long maxnr, long *nr, long rule[],
                  double stat[], int *info);
```

Parameters

1: **rec1** – `long` *Input*

On entry: the index in the data of the first data record used in the analysis.

Constraint: **rec1** ≥ 0.

2: **nrec** – `long` *Input*

On entry: the number of consecutive records, beginning at **rec1**, used in the analysis.

Constraint: **nrec** > 1.

3: **dblk** – `long` *Input*

On entry: the total number of data records (transactions) in the data block.

Constraint: **dblk** $\geq$ **rec1** + **nrec**.

4: **ni** – `long` *Input*

On entry: the highest integer number used to describe an item.

Constraint: **ni** > 1.

5: **nit**[**dblk**] – `long` *Input*

On entry: the number of items in each data record.

Constraint: **nit**$[i] \geq 1$, for $i = 0, 1, \ldots,$ **dblk** $- 1$.

6: **data**$[n]$ – `long` *Input*

On entry: the n items, where n is equal to the sum of elements in **nit**.

7: **supptype** – `long` *Input*

On entry: if **supptype** $= 0$, the support of the antecendent is used to compute support for association rules; otherwise the support of the itemset is used.

Constraint: **supptype** $\in \{0, 1\}$.

8: **minsupp** – `double` *Input*

On entry: the minimum level of support, as a percentage, that each itemset must have to be considered for generating rules.

Constraint: $0.0 \leq$ **minsupp** ≤ 100.0.

9: **minconf** – `double` *Input*

On entry: the minimum level of confidence, as a percentage, in each rule.

Constraint: $0.0 \leq$ **minconf** ≤ 100.0.

10: **maxnir** – `long` *Input*

On entry: the maximum number of items in a rule.

Constraint: $1 <$ **maxnir** $\leq$ **ni**.

11: **maxnr** – `long` *Input*

On entry: **maxnr** is the maximum number of rules to generate.

Constraint: **maxnr** ≥ 1.

12: **nr** – `long *` *Output*

On exit: **nr** the number of generated rules; **nr** $\leq$ **maxnr**.

13: **rule**$[(1 + \textbf{maxnir}) * \textbf{maxnr}]$ – `long` *Output*

On exit: the array **rule** contains the **nr** generated rules. The ith rule contains **rule**$[(1+\textbf{maxnir}) * i]$ items, the last of which is the consequent of the rule, for $i = 0, 1, \ldots, \textbf{nr} - 1$.

14: **stat**$[2 * \textbf{nr}]$ – `double` *Output*

On exit: **stat** contains information on the support and confidence for the **nr** generated rules. For the ith generated rule, **stat**$[i * 2]$ contains the support value and **stat**$[i * 2 + 1]$ the confidence value, for $i = 0, 1, \ldots, \textbf{nr} - 1$.

15: **info** – `int *` *Output*

On exit: **info** gives information on the success of the function call:

0: the function successfully completed its task.

i; $i = 1, 2, \ldots, 5, 7, 8, \ldots, 11$: the specification of the ith formal parameter was incorrect.

99: the function failed to allocate enough memory.

Notation

nrec	the number of data records, n.
nit$[i]$	the number of items in the ith data record, m_i.
data	the item data, X.
minsupp	the minimum support s for a rule.
minconf	the minimum confidence c for a rule.
maxnir	the maximim number r of items in a rule.

Description

The association rules generator **nagdmc_assoc** finds rules in a set X containing n data records x_i, for $i = 1, 2, \ldots, n$. Each data record may contain a different number of integer data values or items. Let the ith record contain a list of m_i items:

$$x_{ij} \in Z^+, \quad j = 1, 2, \ldots, m_i,$$

where $x_{i1} > 0$, and the values are sorted in ascending order:

$$x_{ij} < x_{i(j+1)}, \quad j = 1, 2, \ldots, m_i - 1.$$

In general, let L_k denote a set of k distinct items, known as an itemset, and p be the number of records in X that satisfy:

$$L_k \subset X_i, \quad \text{for } i = 1, 2, \ldots, n.$$

Hence the support of L_k by X is defined as:

$$S(L_k) = \frac{p}{n}.100\%.$$

For a given value s, an itemset with support greater than or equal to s per cent. is known as a frequent itemset.

The algorithm computes the frequent itemsets for itemsets containing from one up to r items, and uses these frequent itemsets:

$$I_k, \quad k = 1, 2, \ldots, r,$$

to compute a set of rules of the kind:

$$\text{Antecendent} \rightarrow \text{Consequent},$$

by forming,

$$\begin{array}{ll} \text{Antecedent:} & A_k = I_k \backslash \{b\}; \\ \text{Consequent:} & C_k = \{b\}; \end{array}$$

where b is the lth item in I_k, for $l = 1, 2, \ldots, k$.

Each rule satisfies conditions for minimum support, s, and confidence, c. Support of a rule can be defined in one of two ways: support of the itemset, $S(I_k)$, as in Agrawal et al. (1993); or support of the items in the antecedent, $S(A_k)$, as in Borgelt and Kruse (2001). Confidence in a rule is defined as:

$$\frac{S(A_k)}{S(I_k)}.100\%.$$

References and Further Reading

Agrawal R, Imielienski T and Swami A (1993) Mining association rules between sets of items in large databases *Proc. Conf. on Management of Data* 207–216 New York: ACM Press.

Borgelt C and Kruse R (2001) Induction of association rules: a priori implementation *Technical Report* School of Computer Science, University of Madgeburg, Germany.

See Also

nagdmc_assoc_data	reads in items from a numeric ASCII file.
nagdmc_assoc_print	prints the rules generated by an association rule analysis.
assoc_ex.c	the example calling program.