Data Imputation: nagdmc_impute_em

Purpose

nagdmc_impute_em imputes missing values in data by using an expectation maximisation (EM) algorithm, assuming a multivariate Normal distribution over m variables.

nagdmc_impute_em returns an array containing the indexes of imputed values (see 'Explanatory Code'). The memory used by this array should be returned to the operating system by the user as indicated in the Essential Introduction.

Declaration

```
#include <nagdmc.h>

long *nagdmc_impute_em(long rec1, long nvar, long nrec, long dblk, double data[],
                       long nxvar, long xvar[], double wt[], double mval,
                       double tol, long maxit, long *it, long *nrepl,
                       long *nempty, double mean[], double cov[], double *sumwts,
                       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: **nvar** – `long` *Input*

On entry: the number of variables in the data.

Constraint: **nvar** ≥ 1.

3: **nrec** – `long` *Input*

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

Constraint: **nrec** > 1.

4: **dblk** – `long` *Input*

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

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

5: **data**[**dblk** $*$ **nvar**] – `double` *Input/Output*

On entry: data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar** $- 1$) are stored in **data**$[i *$ **nvar** $+ j]$, for $i = 0, 1, \ldots,$ **dblk** $- 1$.

On exit: missing values in **data** are replaced by their estimates.

6: **nxvar** – `long` *Input*

On entry: the number of variables in the analysis. If **nxvar** $= 0$, all variables in the data are used in the analysis.

Constraint: $0 \leq$ **nxvar** $\leq$ **nvar**.

7: **xvar**[**nxvar**] – `long` *Input*

On entry: the indices indicating the position in **data** in which the variables in the analysis are stored. If **nxvar** $= 0$ then **xvar** must be 0, and the indices of variables are given by $j = 0, 1, \ldots,$ **nvar** $- 1$.

Constraints: if **nxvar** > 0, $0 \leq$ **xvar**$[i] <$ **nvar**, for $i = 0, 1, \ldots,$ **nxvar** $- 1$; otherwise **xvar** must be 0.

8: **wt**[**dblk**] – `double` *Input*

On entry: **wt**$[i]$ contains the weight on the ith data record, for $i = 0, 1, \ldots,$ **dblk** $- 1$. If the weight on each record is the same, **wt** should be set equal to 0.

Constraint: if **wt** $\neq 0$, **wt**$[i] \geq 0.0$, for $i = 0, 1, \ldots,$ **dblk** $- 1$.

9: **mval** – `double` *Input*

On entry: all values in **data** equal within machine precision to **mval** are considered missing from the analysis.

Suggested value: a value outside the interval $[a, b]$, where a and b are the minimum and maximum value in your data, respectively.

10: **tol** – `double` *Input*

On entry: the convergence tolerance of the EM algorithm.

Constraint: **tol** > 0.0.

11: **maxit** – `long` *Input*

On entry: the maximum number of iterations of the EM algorithm.

Constraint: **maxit** ≥ 1.

12: **it** – `long` *Output*

On exit: the actual number of iterations performed by the EM algorithm.

13: **nrepl** – `long *` *Output*

On exit: the number of imputed values in **data**, and equals the number of values in **data** equal to **mval**.

14: **nempty** – `long *` *Output*

On exit: the number of data records for which the m variables in the analysis each take the value **mval**.

15: **mean**$[m]$ – `double` *Output*

On exit: if **mean** $\neq 0$, **mean**$[i]$ contains the mean value of the ith variable in the analysis after imputation, for $i = 0, 1, \ldots, m - 1$; otherwise **mean** is not referenced.

16: **cov**$[m * (m + 1)/2]$ – `double` *Output*

On exit: if **cov** $\neq 0$, the first $m * (m + 1)/2$ elements of **cov** contain the upper-triangular part of the variance-covariance matrix of the m variables in the analysis, packed by row; otherwise **cov** is not referenced.

17: **sumwts** – `double *` *Output*

On exit: the sum of the case weights in the model. If **wt** is set to 0, **sumwts** equals the number of data records.

18: **info** – `int *` *Output*

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

0: the function successfully completed its task.

-20: the EM algorithm has not converged and all results should be treated with caution. The value of **maxit** should be increased before executing the function again. If the EM algorithm persists in having difficulty converging, a lower value of **tol** should be considered.

i; $i = 1, 2, 3, 4, 6, 7, 8, 10, 11$: the specification of the ith formal parameter was incorrect.

19: covariance matrix not positive definite.

20: no missing values were found in the data; check your definition of **mval**.

99: the function failed to allocate enough memory.

Notation

nrec the number of data records, n.
data the data set X.
nxvar determines the number of variables, m.
mval the value of missing data values, z.
tol the tolerance for terminating the EM algorithm, τ.
nrepl the number of replaced values, r.
mean the estimated sample means, $\hat{s}$.
cov the estimated sample variance-covariances, $\hat{\Sigma}$.

Description

The EM algorithm is used to estimate the mean and variance-covariance matrix of a multivariate Normal probability distribution given a set, X, of n data records with m variables and $r > 0$ values missing at random.

Let z be the value of a real scalar used to identify values missing at random in the data. Excluding data values equal to z, the first step is to compute initial estimates of the sample means $\widehat{s}$ and variance-covariance matrix $\widehat{\Sigma}$. In each iteration of the EM algorithm the estimates $\widehat{s}$ and $\widehat{\Sigma}$ are revised in three steps.

Firstly, given a data record x_i with $p < m$ known values and $q = m - p$ missing values, let $\widehat{\Sigma}_{kk} \in \mathbb{R}^{p\times p}$ be the estimated variance-covariance matrix over the variables with known values and $\widehat{\Sigma}_{km} \in \mathbb{R}^{p\times q}$ be the estimated variance-covariance matrix of the variables with known values and the variables with missing values, both of which are obtained from $\widehat{\Sigma}$. Then the linear regression of variables with missing values on variables with known values has coefficients B given by,

$$B = \widehat{\Sigma}_{kk}^{-1}\widehat{\Sigma}_{km}.$$

Secondly, the q missing values in x_i are replaced by their conditional expectation values given the p known values and the estimates of $\widehat{s}$ and $\widehat{\Sigma}$ by using the regression model:

$$u_u = v_u + (u_k + v_k)B,$$

where

$u_u \in \mathbb{R}^{1\times q}$ contains the estimates of the q missing values,
$v_u \in \mathbb{R}^{1\times q}$ contains the estimated sample means from $\widehat{s}$ for the variables with missing values,
$u_k \in \mathbb{R}^{1\times p}$ contains the data values from x_i for the variables with known values,
and $v_k \in \mathbb{R}^{1\times p}$ contains the estimated sample means from $\widehat{s}$ for the variables with known values.

Finally, as each data record containing at least one missing value is updated, the revised estimates of the sample means and variance-covariance matrix are computed by using West's update algorithm.

At the end of the jth iteration of the EM algorithm, the largest value, say $w \in \mathbb{R}$, of the absolute values of the differences $\widehat{s}^{(j-1)} - \widehat{s}^{(j)}$ and $\widehat{\Sigma}^{(j-1)} - \widehat{\Sigma}^{(j)}$ is compared against a tolerance τ and the EM algorithm terminates if $w \leq \tau$.

References and Further Reading

Dempster A P N, Laird N M and Rubin D B (1977) Maximum likelihood estimation from incomplete data via the EM algorithm *J. Roy. Stat. Soc. B* **39** 1–38.

Little R J A and Rubin D B (1987) *Statistical Analysis with Missing Data* Wiley.

West D H D (1979) Updating mean and variance estimates: an improved method *Comm. ACM* **22** (9) 532–535.

Explanatory Code

The following C function prints the index in the data and the imputed value of each of the **nrepl** replaced values after a successful call to **nagdmc_impute_em** returning `indexes`.

```
#include <stdio.h>

void imputed_values(long nvar, double data[], long nrepl, long indexes[]) {
    long i, j, k;

#define MISSING_ROW(I) indexes[I]
#define MISSING_COL(I) indexes[I+nrepl]
#define DATA(I,J) data[(I)*nvar+J]

    printf("\n\tRow \tCol \tValue\n");
    for (i=0; i<nrepl; ++i) {
        j = MISSING_ROW(i);
        k = MISSING_COL(i);
        printf("\t%-4li\t%-4li\t%-8.4f\n",j,k,DATA(j,k));
    }
}
```

See Also

nagdmc_free_impute	returns memory allocated by **nagdmc_impute_em** to the operating system.
impute_em_ex.c	the example calling program.