Turku PET Centre Modelling report TPCMOD0020 2003-05-23
Kaisa Sederholm

Using NNLS in multilinear PET problems

This document reviews the nonnegative least square (NNLS) optimisation problem and the usage of the NNLS algorithm [Lawson & Hanson 1974] in multilinear problems. NNLS algorithm is programmed in C language and it can be found from the library libpet. Verifications of the accuracy of the algorithm are presented in Appendix A.

Problem NNLS

By nonnegative least square problem we mean a linear programming problem that can be expressed in the following way

$$\min\|Ex - f\|,\ x \geq 0\ .$$

Thus this is a least square problem, where the only constraints are the nonnegativity conditions. In linearized PET-models we can usually constrain the macroparameters to be nonnegative and thus end up with an NNLS problem.

The NNLS algorithm

Basic idea

The basic idea of NNLS algorithm is based on *duality rules* of linear programming [Lawson & Hanson 1974]. If the least square solution vector x satisfies

$$x_j > 0 \quad j \in P,$$
$$x_j = 0 \quad j \in Z \text{ and}$$
$$Ex \cong f$$

then the dual vector $w = E^T(f - Ex)$ satisfies conditions

$$w_j = 0 \quad j \in P,$$
$$w_j \leq 0 \quad j \in Z \quad \text{and}$$
$$w = E^T(f - Ex).$$

The NNLS algorithm tries to find a vector y that satisfies the dual conditions. The dual problem has a solution if and only if the original problem has a solution. At first the dual vector is computed based on solution $x=0$. Based on vector w, we decide which parameter value we start to improve. The flow chart of the algorithm is presented below. Variables indexed in the set P are free to take values different from zero and variables indexed in set Z are held at the value zero.

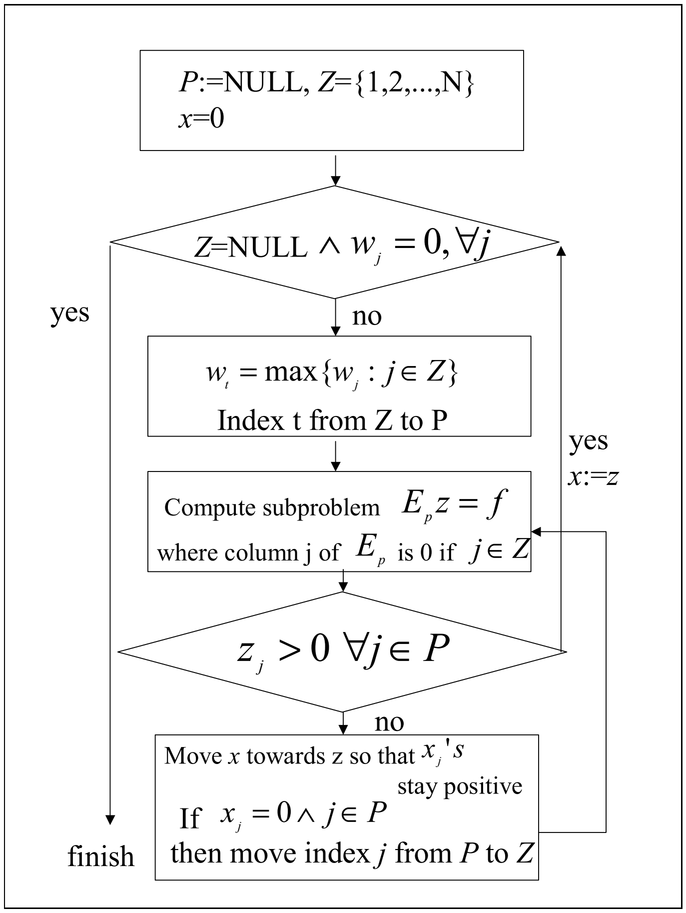

Figure 1. Flow chart of NNLS.

Input/Output of the algorithm

The parameters NNLS algorithm needs are:

1. The $m \times n$ matrix *E* in the array *a[n][m]*
2. Value of *m*
3. Value of *n*
4. Vector f in *m*-array *b[m]*
5. Empty array *x[n]* for the solution
6. Empty variable *rnorm* for the Euclidean norm of the residual vector
7. An n-array of working space *wp[n]* for the dual solution *y*
8. An m-array of working space *zzp[m]*
9. An n-array of working space *indexp[n]*

Algorithm generates an $m \times m$ orthogonal matrix *Q*, that satisfies the condition *QE*=*R*, where *R* is $n \times n$ upper triangular matrix. At the end of the algorithm, array *a[n][m]* contains the product matrix *QE*, and array *b[m]* contains vector Qf . Norm of the residual vector is saved in variable *rnorm*. The solution of the least square problem is in vector *x*, and the dual solution in vector *wp*.

Multilinear problems

In PET-modelling the problems are usually nonlinear. However, we have the means to transform a nonlinear problem into a linear one [G. Blomquist 1984].
Let's consider the five-parameter FDG model [W. Cai et al. 2002]

$$\begin{cases} \frac{d}{dt}C_e^*(t) = k_1^* C_p^*(t) - (k_2^* + k_3^*)C_e^*(t) + k_4^* C_m^*(t) \\ \frac{d}{dt}C_m^*(t) = k_3^* C_e^*(t) - k_4^* C_m^*(t) \end{cases},$$

where the total concentration in tissue is given by $C_i^*(t) = C_e^*(t) + C_m^*(t)$. By introducing the macroparameters

$$\begin{aligned} &P_1 = CBV, \quad P_2 = (k_2^* + k_3^* + k_4^* - k_1^*) \cdot CBV + k_1^* \\ &P_4 = -(k_2^* + k_3^* + k_4^*), \quad P_5 = -(k_2^* k_4^*) \\ &P_3 = (k_2^* k_4^* - k_1^* k_3^* - k_1^* k_4^*) \cdot CBV + k_1^*(k_3^* + k_4^*) \end{aligned} \tag{1}$$

after several operation [W. Cai et al. 2002] a linearized form of the FDG model can be obtained.

$$C_i^*(t) = P_1 C_p^*(t) + P_2 \int_0^t C_p^*(\tau)d\tau + P_3 \int_0^t \int_0^\tau C_p^*(s)ds d\tau + P_4 \int_0^t C_i^*(\tau)d\tau + P_5 \int_0^t \int_0^\tau C_i^*(s)ds d\tau$$

When $f = [C_i^*(t_1), C_i^*(t_2), ..., C_i^*(t_m)]^T$ and $x = [P_1, P_2, ..., P_5]^T$, we end up with a least square problem $\min\|Ex - f\|, x \geq 0$. The coefficient matrix is constructed so that it contains the values of the coefficient at each time point as columns. Thus in this case

$$E = \begin{pmatrix} C_p^*(t_1) & \int_0^{t_1} C_p^*(\tau)d\tau & \int_0^{t_1}\int_0^\tau C_p^*(s)ds d\tau & \int_0^{t_1} C_i^*(\tau)d\tau & \int_0^{t_1}\int_0^\tau C_i^*(s)ds d\tau \\ C_p^*(t_2) & \int_0^{t_2} C_p^*(\tau)d\tau & \int_0^{t_2}\int_0^\tau C_p^*(s)ds d\tau & \int_0^{t_2} C_i^*(\tau)d\tau & \int_0^{t_2}\int_0^\tau C_i^*(s)ds d\tau \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C_p^*(t_m) & \int_0^{t_m} C_p^*(\tau)d\tau & \int_0^{t_m}\int_0^\tau C_p^*(s)ds d\tau & \int_0^{t_m} C_i^*(\tau)d\tau & \int_0^{t_m}\int_0^\tau C_i^*(s)ds d\tau \end{pmatrix}.$$

When the macroparameters $P_1, ..., P_5$ are constrained to be nonnegative, this linear optimisation problem can be given for the NNLS algorithm. Coefficient matrix *E* is given in array *a[n][m]* and vector f is given in array *b[m]*. Number of the macroparameters is given as the value of *n* and the number of timepoints as the value of m. When the algorithm has finished, values of the macroparameters can be read from the array *x[n]* and values of the microparameters k_1, k_2, k_3 and k_4 can then be calculated based on equations (1).

References

1. Lawson, C. L., Hanson, R.J.: *Solving Least Squares Problems,* Prentice-Hall, Englewood Cliffs, New Jersey, 1974
2. Blomqvist G.: On the construction of functional maps in positron emission tomography. *Journal of Cerebral Blood Flow and Metabolism* 1984;4:629-632.
3. Cai, W., Feng, D., Fulton, R., Siu, W.C.:Generalized linear least squares algorithms for modelling glugose metabolism in the human brain with correction for vascular effects, *Computer Methods and Programs in Biomedicine* 68 (2002) 1-14