Turku PET Centre Modelling report TPCMOD0025 2003-09-11
Kaisa Sederholm

Using QR decomposition for overdetermined systems

This document reviews least square optimisation problem with no constraints (see modelling report TPCMOD0020 for least square problem with nonnegativity constraints) and the usage of QR and Householder algorithms [Lawson & Hanson 1974] [Heath 2002] [Householder 1957]. Both algorithms have been programmed in C language and can be found in the library libpet. Verifications of accuracy of the algorithms are presented in Appendix A.

Overdetermined least square problem

The matrix form of an overdetermined least square problem is

$$\min\|Ax - b\|,$$

where A is m×n (m>n) coefficient matrix and vector b can be considered as the data vector. The solution of this problem is vector x^*, that minimizes the Euclidean norm.

QR algorithm

Idea of the QR algorithm is to transform coefficient matrix A into product of two matrixes Q and R [Lawson & Hanson 1974] [Heath 2002]. Then the problem can be expressed with equation

$$QRx \cong b,$$

where Q is an orthogonal m×m matrix and R is upper triangular m×n matrix. Since for orthogonal matrixes $Q^{-1}=Q^T$, the equation can be transformed into form

$$Rx \cong Q^T b.$$

After this it is easy to inverse the upper triangular matrix and get the solution from the equation

$$x \cong R^{-1}(Q^T b).$$

Householder algorithm

Householder algorithm [Householder 1957] [Lawson & Hanson 1974] [Heath 2002] works as a subroutine that QR algorithm uses to form the matrix decomposition QR.

By Householder transform we mean a transformation matrix that has form

$$H = I - 2\frac{uu^T}{u^T u},$$

where u is a nonzero vector. Transformation changes an arbitrary vector v into a multiple of a unit vector.

$$Hv = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \alpha\, e_1$$

We get the transformation vector u from the formula

$$u = v - \alpha\, e_1, \quad \alpha = \pm\|v\|$$

There is no need to save the whole matrix H – vector u and constant α define the transformation.

Householder QR factorisation

Transforming repetitively each column of A we get upper triangular matrix

$$R = H_n H_{n-1} \cdots H_1 A.$$

Matrix H_1 is the transformation matrix of the first column of A, matrix H_2 the transformation matrix for the last n-1 elements of the second column and so on. Product $H_1 H_2 \ldots H_n$ forms the orthogonal matrix Q.

Input/Output of qr.c

The parameters needed in the call of qr.c algorithm of library libpet are:

1. The $m \times n$ matrix A in the array *A[m][n]*
2. Value of m
3. Value of n
4. Vector b 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 $r=Ax-b$
7. An n-array of working space *tau[n]*.
8. An m-array of working space *res[m]*.
9. An m×n array of working space *wws[m][n]*
10. A 2m-array of working space *ws[2m]*

At the end of the algorithm, upper triangular part of array *a[n][m]* contains matrix R and lower triangular part contains Householder vectors $u_1 \ldots u_n$. Householder constants $\alpha_1 \ldots \alpha_n$ are saved in vector *tau*. Array *b[m]* will remain untouched. Norm of the residual vector is saved in variable *rnorm*. The solution of the least square problem is given in vector *x* and the residual vector in vector *res*. A null-pointer can be given for the last four variables – then the memory will be allocated within qr.c.

References

1. Lawson, C. L., Hanson, R.J.: *Solving Least Squares Problems,* Prentice-Hall, Englewood Cliffs, New Jersey, 1974
2. Heath, Michael T.: *Scientific computing: An introductory survey, Second Edition*, McGraw-Hill, New York, 2002.
3. Householder A. S.: *The Approximate solution of Matrix Problems*. Oak Ridge National Laboratory, Oak Ridge, Tennesee, 1957.