CAAM420 Lecture Notes

Chang Da

November 22nd, 2013

1 Announcements

The last project is generally based on project 6 we have just worked on. There won't be much programming. The main part is already done. This makes project 6 quite important because it puts nearly everything in this course together. Thus it is highly recommanded to continue working on this project and ask the instructor or other students about any confusion since it is not pledged.

Late submission WILL be accepted only for this assignment. Although you will get some points of every day after the due date.

If you can get this project under control, there will be very little significant work to do for the next pledged project.

2 Least Square Problem

We want to solve $Ax = b$, in which A is a m-by-n matrix and m may not be equal to n.

Converting it into a least square problem, what we actually need to solve becomes: minimizing $(Ax - b)^T(Ax - b)$, in which $(Ax - b)^T$ is called the residual vector.

On Monday we proved that the solution x is a solution of the so-called normal equation $A^TAx = A^Tb$.In the original equation, b may not be in the column space of A; but in the normal equation it is guaranteed. And this normal equation is a square linear system and can be simply solved by Gaussian Elimination.

However, least square problems are better approached through understanding the column space of a matrix, which is referred to as Typical Dense Matrix Algorithm.In this algorithm we use the QR decomposition: $A = QR$, in which A is an m-by-n matrix; Q is an orthorgonal and normalized, or orthonormal m-by-n matrix, $Q^TQ = I$; R is an upper triangular n-by-n matrix.

The beauty of QR decomposition is that the column space of A equals the column space of Q. Thus, x solves the least square problem <==> $Ax - b$ is propendicular to the column space of A.

If A has a full rank(A^TA is invertible), then R is invertible, with $Q^TQ = I$, and our solution is $x = R^{-1}Q^Tb$.

In the CLAPACK, we use *dgels and *sgels functions to do QR decomposition and solve least square problems. Let's take a look at some detailed comments in sgels function.

3 Comment from sgels.c

```
/*  SGELS solves overdetermined or underdetermined real linear systems */
/*  involving an M-by-N matrix A, or its transpose, using a QR or LQ */
/*  factorization of A.  It is assumed that A has full rank. */

/*  The following options are provided: */

/*  1. If TRANS = 'N' and m >= n:  find the least squares solution of */
/*     an overdetermined system, i.e., solve the least squares problem */
```

```
/*                  minimize || B - A*X ||. */

/*  2. If TRANS = 'N' and m < n:  find the minimum norm solution of */
/*     an underdetermined system A * X = B. */

/*  3. If TRANS = 'T' and m >= n:  find the minimum norm solution of */
/*     an undetermined system A**T * X = B. */

/*  4. If TRANS = 'T' and m < n:  find the least squares solution of */
/*     an overdetermined system, i.e., solve the least squares problem */
/*                  minimize || B - A**T * X ||. */

/*  Several right hand side vectors b and solution vectors x can be */
/*  handled in a single call; they are stored as the columns of the */
/*  M-by-NRHS right hand side matrix B and the N-by-NRHS solution */
/*  matrix X. */

/*  Arguments */
/*  ========= */

/*  TRANS   (input) CHARACTER*1 */
/*          = 'N': the linear system involves A; */
/*          = 'T': the linear system involves A**T. */

/*  M       (input) INTEGER */
/*          The number of rows of the matrix A.  M >= 0. */

/*  N       (input) INTEGER */
/*          The number of columns of the matrix A.  N >= 0. */

/*  NRHS    (input) INTEGER */
/*          The number of right hand sides, i.e., the number of */
/*          columns of the matrices B and X. NRHS >=0. */

/*  A       (input/output) REAL array, dimension (LDA,N) */
/*          On entry, the M-by-N matrix A. */
/*          On exit, */
/*            if M >= N, A is overwritten by details of its QR */
```

```
/*                          factorization as returned by SGEQRF; */
/*             if M <  N, A is overwritten by details of its LQ */
/*                          factorization as returned by SGELQF. */

/*  LDA     (input) INTEGER */
/*          The leading dimension of the array A.  LDA >= max(1,M). */
```

In this function it will decide whether to use A or A^T, and we don't have to call the transpose by ourselves.

4 Regularization

First pay attention to ls.c under src files which is quite similar to what will be asked in the next project.

One classic problem is that lots of information blurring in optimization refers to the phenomenon of averaging. Average kills high frequency information. Information has been lost during the process of cancelling and averaging the signals, we cannot get them back. That's why we need regularization by supplying more information to the system.

The simplest form of regularization is just: adding another $\lambda Ix = 0$ to the original system $Ax = b$ This ask the solution to be short.

Without regularization the solution will not be accurate.

We will talk more about regularization next Monday.