Factoring Symmetric Indefinite Matrices

Voicu Chis

April 17, 2007

Motivation. IPM requires solving:

- $$\begin{bmatrix} 0_m & A & 0_{m,n} \\ A^T & 0_n & I_n \\ 0_{n,m} & S & X \end{bmatrix} \begin{bmatrix} \triangle\lambda \\ \triangle x \\ \triangle s \end{bmatrix} = b$$
  $A \in M_{m,n}(\mathbb{R})$ full rank, $X, S \in M_n(\mathbb{R})$ diagonal and PD
  I identity matrix, size n
  $\triangle\lambda \in \mathbb{R}^m, \triangle x \in \mathbb{R}^n, \triangle s \in \mathbb{R}^n, b \in \mathbb{R}^{m+2n}$
  the matrix of the system is a $2n + m$ square matrix
- equivalently:
  $\begin{bmatrix} 0_m & A \\ A^T & -D^{-2} \end{bmatrix} \begin{bmatrix} \triangle\lambda \\ \triangle x \end{bmatrix} = b^*$ (Augmented System (AS))
  $\triangle s = -X^{-1}(b^{**} + S \triangle x)$
  $D = S^{-1/2}X^{1/2} \in M_n(\mathbb{R})\ \text{diagonal}\ \ \text{and}\ PD, b^* \in \mathbb{R}^{m+n}$
  the matrix of the system is a $n + m$ square matrix
  the matrix is symmetric but **indefinite**
- equivalently the equations in Yang's introduction that reduces to solving:
  $AD^2A^T \triangle \lambda = b^{***}$, a system with a $n \times n$ PD symmetric matrix

Why solving the AS instead of normal equations?

- the dense columns of A don't destroy the sparsity of the augmented system matrix as they were destroying the sparsity of the matrix in the normal equations-remember Yang's explanation
- ill conditioning occurs in both systems but in the augmented one it is easier to trace the effects on the numerical factorization
- in Quadratic Programming we must solve either a system having the matrix $A^T(D^2+Q)^{-1}A$ or one having the matrix $\begin{bmatrix} Q & A \\ A^T & -D^2+Q \end{bmatrix}$. It is worth working with the second system because the inverse in the first matrix makes it very dense.

Why solving the normal equations instead of the AS?

- one is bigger than the other. More computer time needed.
- "algorithms and software for solving **sparse** symmetric indefinite systems are not as highly developed or available as sparse Cholesky code"-Stephen J. Wright, 1997

Remark. Symmetric indefinite systems must be solved when dealing with finite element methods for PDEs, nonlinear optimization, electrical network modelling.

Our goal. Solve 'smartly' $\bar{A}x = b$ when $\bar{A}$ is symmetric but **indefinite**.
'Smart:'

- think about a 'useful' factorization of $\bar{A}$ that always exists and can be numerically stable obtained
- use the symmetry of $\bar{A}$ to do 'half' work as in Gaussian elimination
- use the sparsity of $\bar{A}$ when factorizing
- do the factorization in a numerical stable way

In IPM, note:

- at each step the diagonal scaling matrix is the only thing changing in the AS matrix. This gives a **constant sparsity structure**.

How to factorize Symmetric Indefinited Matrices:

- If $\bar{A}$ is symmetric but indefinite there might not exist a permutation matrix P such that $P\bar{A}P^T$ can be LU factorized, i.e. $P\bar{A}P^T = LDL^T$ for some L unit lower triangular and diagonal D. Example. $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ (see Gill&Murray&Wright [1])
- Even when exists such P we have numerical stability problems (can not control the growth) $\begin{pmatrix} \epsilon & 1 \\ 1 & \epsilon \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{pmatrix} \begin{pmatrix} \epsilon & 0 \\ 0 & \epsilon - 1/\epsilon \end{pmatrix} \begin{pmatrix} 1 & 1/\epsilon \\ 0 & 1 \end{pmatrix}$ (see Gill&Murray&Wright [1])
- Cholesky factorization is LDL^T factorization with $D = I$
- What to do?
  Remark. If $\bar{A}$ is symmetric indefinite then a symmetric permutation P exists such that either the first element in $P\bar{A}P^T$ is nonzero or the upper left 2×2 block of $P\bar{A}P^T$ is nonsingular.
  Remark. This guarantees the existence of the factorization $P\bar{A}P^T = LBL^T$ where P is a permutation, L is unit lower triangular, B is symmetric and block diagonal with 1×1 or 2×2 blocks.

In the previous slide we said:

Remark. If $\bar{A}$ is symmetric indefinite then a symmetric permutation P exists such that either the first element in $P\bar{A}P^T$ is nonzero or the upper left 2x2 block of $P\bar{A}P^T$ is nonsingular.

More details. If P is a permutation matrix then

- $P\bar{A}P^T$ and $\bar{A}$ have the same elements on the diagonal, just scrambled.
- the upper left 2×2 corner of $P\bar{A}P^T$ (which is symmetric) has on the main diagonal 2 elements from the diagonal of $\bar{A}$ and the other element in the block is some off-diagonal element of matrix $\bar{A}$
- in other words, given any 2 elements on the main diagonal of $\bar{A}$ $(a_{ii}, a_{jj}, i \neq j)$ **there exist a unique** permutation matrix P such that the upper left 2x2 block in $P\bar{A}P^T$ is $\begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix}$

- **Step 1.** We fix a permutation matrix P that either brings a non-zero element or a 2x2 non-singular block in the upper left corner of $P\bar{A}P^T$. The element or the block is refered to as 'the pivot'. This operation is refered to as 'choosing the pivot'
- **Step 2.** We write $P\bar{A}P^T = \begin{bmatrix} E & C^T \\ C & T \end{bmatrix}$ where E is our pivot which makes C either $(n-1)\times 1$ or $(n-2)\times 2$ and T is what is left. We can see that
$$P\bar{A}P^T = \begin{bmatrix} I & 0 \\ CE^{-1} & I \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & T - CE^{-1}C^T \end{bmatrix} \begin{bmatrix} I & 0 \\ CE^{-1} & I \end{bmatrix}^T$$
Note. CE^{-1} is either $(n-1)\times 1$ or $(n-2)\times 2$ giving us the first column respectively the first 2 columns (namely $\begin{bmatrix} I \\ CE^{-1} \end{bmatrix}$) in our L from the factorization $P\bar{A}P^T = LDL^T$. Also E is the first block in D.
- **Step 3.** Note that $T - CE^{-1}C^T$ is symmetric and indefinite but of dimension $n-1\times n-1$ or $n-2\times n-2$. We repeat Step 1 and Step 2 to this matrix adding columns (1 or 2) to L and another block to D

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

1. Dealing with numerical stability

Remember. The added colon(s) in L and the new matrix. Numerical stability means controling the growth of elements in CE^{-1} and $T - CE^{-1}C^T$.

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

1. Dealing with numerical stability

Remember. The added colon(s) in L and the new matrix. Numerical stability means controling the growth of elements in CE^{-1} and $T - CE^{-1}C^T$.

Bunch-Parlett algorithm-Complete Pivoting

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

1. Dealing with numerical stability

Remember. The added colon(s) in L and the new matrix. Numerical stability means controling the growth of elements in CE^{-1} and $T - CE^{-1}C^T$.

Bunch-Parlett algorithm-Complete Pivoting

- Examines all elements in $\bar{A}$ (the new $\bar{A}$ at each step (remaining matrix)) and identify the largest diagonal element, denoted a_{kk}, and the largest off-diagonal element, denoted a_{ij}.

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

1. Dealing with numerical stability

Remember. The added colon(s) in L and the new matrix. Numerical stability means controling the growth of elements in CE^{-1} and $T - CE^{-1}C^T$.

Bunch-Parlett algorithm-Complete Pivoting

- Examines all elements in $\bar{A}$ (the new $\bar{A}$ at each step (remaining matrix)) and identify the largest diagonal element, denoted a_{kk}, and the largest off-diagonal element, denoted a_{ij}.
- The prefered 1×1 pivot is a_{kk}

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

1. Dealing with numerical stability

Remember. The added colon(s) in L and the new matrix. Numerical stability means controling the growth of elements in CE^{-1} and $T - CE^{-1}C^T$.

Bunch-Parlett algorithm-Complete Pivoting

- Examines all elements in $\bar{A}$ (the new $\bar{A}$ at each step (remaining matrix)) and identify the largest diagonal element, denoted a_{kk}, and the largest off-diagonal element, denoted a_{ij}.
- The prefered 1×1 pivot is a_{kk}
- The prefered 2×2 pivot is $\begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix}$

Remark in the previous algorithm

Freedom to choose PIVOTS
Keep in mind: numerical stability and sparsity.

1. Dealing with numerical stability

Remember. The added colon(s) in L and the new matrix. Numerical stability means controling the growth of elements in CE^{-1} and $T - CE^{-1}C^T$.

Bunch-Parlett algorithm-Complete Pivoting

- Examines all elements in $\bar{A}$ (the new $\bar{A}$ at each step (remaining matrix)) and identify the largest diagonal element, denoted a_{kk}, and the largest off-diagonal element, denoted a_{ij}.
- The prefered 1×1 pivot is a_{kk}
- The prefered 2×2 pivot is $\begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix}$
- a rigorous way and analysis for choosing 2×2 pivot instead of 1×1 pivot and the bound for growth can be found p.117 [1]-**modest growth**

Bunch-Kaufman algorithm-Partial Pivoting

- Selecting the pivot comes after analysing just 2 columns of the remaining matrix. **Lower cost**
- See algorithm description pg. 119 in [1]

2. Dealing with sparsity

Goal. Not create to much fill-in in the remaining matrix, $T - CE^{-1}C^T$.

- Fourer&Mehrotra [2] modify the Bunch-Parlett strategy adding a sparsity preserving criterion
- they count the number of zeros in $T - CE^{-1}C^T$ for all 1×1 and 2×2 pivots
- we start with the pivot producing the maximum number of zeros and go down until a certain stability condition is checked.

References:

1. P. Gill, W. Murray, M.H. Wright - Numerical Linear Algebra and Optimization (1991)
2. R. Fourer, S. Mehrotra - Solving symmetric indefinite systems in an interior-point method for linear programming, Mathematical Programming, 62 (1993)
3. Stephen J. Wright: "Prima-Dual Interior-Point Method"
4. Andersen E.D., J. Gondzio, C. Meszaros, and X. Xu, Implementation of Interior Point Methods for Large Scale Linear Programming, in: Interior Point Methods in Mathematical Programming, T. Terlaky (ed.), Chapter 6, pp. 189-252, Kluwer Academic Publisher, 1996 -can be found on the web.
5. Vanderbei R., Carpenter T. - Symmetric Indefinite Systems for IPM, Mathematical Programming 58 (1993), 1-32
6. J.R. Bunch, B.N. Parlett - Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations, SIAM Journal on Numerical Analysis 8 (1971), 639-655