Christopher Wong
999234204
MAT 128B

Programming Project 2

2/15/2017

(1) Show that A is nonsingular.

We are given the definition that a matrix $A \in \mathbb{R}^{n\times n}$ is said to be symmetric positive definite if A is symmetric, i.e., $A = A^T$, and it satisfies

$$x^T Ax > 0 \quad \text{for all} \quad x \in \mathbb{R}^n, \quad x \neq 0.$$

Recall that a matrix is nonsingular if $Ax = 0$ has no non-trivial solution and that a symmetric positive definite matrix has real positive eigenvalues. So we have that $\lambda \neq 0$ which implies that the equation $Ax = \lambda x = 0$ has a solution if and only if $x = 0$ which is a trivial solution. Thus, A must be nonsingular.

In case it is not obvious that a (symmetric) positive definite matrix has real positive eigenvalues, recall that eigenvalues satisfy the equation $Ax = \lambda x$. Then $x^T Ax = x^T \lambda x = \lambda x^T x > 0$ because λ is a constant and $x^T x$ is the inner product of x with itself. The only way $\lambda x^T x$ is positive is when $\lambda > 0$. Since λ was arbitrarily chosen, we can say that all eigenvalues of a (symmetric) positive definite matrix are positive.

(2) Show that the diagonal entries of A are all positive.

Let A be a symmetric positive definite matrix. Consider a nonzero vector x with a 1 in its j-th position. Then $x^T Ax$ equals the j-th diagonal element of A (a_{jj}). Since $x^T Ax > 0$ for all nonzero $x \in \mathbb{R}^n$ we have that $a_{jj} > 0$. Since our choice of j was arbitrary, we can show that $a_{jj} > 0$ for all $j = 1, 2, ..., n$.

(3) Let $M \in \mathbb{R}^{n\times x}$ be nonsingular. Show that the matrix

$$M^T AM$$

is symmetric positive definite.

We want to show that $x^T M^T AMx > 0$ for all nonzero $x \in \mathbb{R}^n$. Consider the eigenvalues of $M^T AM$ so we get $M^T AMx = \lambda_{(M^T AM)}x$. Similar to part (1), we will show that the eigenvalues of $M^T AM$ are positive and thus proving it is positive definite. If we consider

$$Mx = \lambda_M x$$
$$M^T x = \lambda_{M^T} x$$

we can show that the eigenvalues λ_M and λ_{M^T} are the roots of the same characteristic polynomial because $det(M - \lambda I) = det((M - \lambda I)^T) = det(M^T - \lambda I)$ since $det(M) = det(M^T)$. Then $\lambda_M \lambda_{M^T} = \lambda_M^2 > 0$. From part (1) we also have that $\lambda_A > 0$ so we get that $\lambda_{(M^T AM)} = \lambda_{M^T}\lambda_A\lambda_M > 0$. So we have that $x^T M^T AMx = x^T \lambda M^T AMx = \lambda_{(M^T AM)} x^T x > 0$. Thus, $M^T AM$ is positive definite. Showing that it is symmetric is as easy as showing $(M^T AM)^T = M^T A^T M = M^T AM$ since A is symmetric by definition. Thus, $M^T AM$ is symmetric positive definite.

(B) Show that all pivots arising in LU factorization (without pivoting) applied to A are positive.

In LU factorization without pivoting, the pivots are stored in the diagonals of U. So it suffices to show that all diagonal entries of U are positive. Recall that the determinant of the product of two square matrices of equal size is equal to the product of their determinants. That is, $det(A) = det(LU) = det(L)det(U)$. Furthermore, the determinant of a triangular matrix is the product of its diagonal entries. Since L has 1's on its diagonal, we have $det(L) = 1$ and the diagonals of U are the pivots. So we have that $det(A) = det(U)$. But $det(A) = \prod_{i=1}^{n} \lambda_i$ where λ_i's are the eigenvalues of A.

An additional fact we need is that principal submatrices of a positive definite matrix are also positive definite. To see this, set the vector $x \in \mathbb{R}^n$ accordingly as in part (2). That is, let $x_1, x_2, ..., x_k = 1$ for $k \leq n$ and the remaining entries equal to 0 to show the $k \times k$ principal submatrix is positive definite.

So we now have that the diagonal elements of A (a_{jj}) are positive. It follows then that the diagonal elements of U (u_{jj}) are positive as well since $det(A) = det(U)$. But the diagonals of U are the pivots arising from LU factorization! Thus, all pivots arising in LU factorization (without pivoting) applied to A are positive.

(4) Show that the resulting factorization can be rewritten in the form

$$A = LDL^T$$

where $L \in \mathbb{R}^{n\times n}$ is a unit lower-triangular matrix and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive diagonal entries.

Going through LU factorization, we find that the pivots are stored in the diagonal of U. We will create a new diagonal matrix D using the diagonal elements of U. Notice that multiplying a matrix on the left by a diagonal matrix D multiplies the corresponding row of U by the diagonal element. That is, the i^{th} row of DU is given by d_{ii} * (i^{th} row of U). So if we pull the diagonal elements of U to create the diagonal matrix D (and thus leave 1's on the diagonal of U), we would have to divide the remaining row elements by its pivot. That is, $u'_{ij} = \frac{u_{ij}}{u_{ii}}$ for $j = i+1, i+2, ..., n$ and $d_{ii} = u_{ii}$ and $u'_{ii} = 1$.

Now that we have the decomposition $A = LDU$, we can see that $A^T = (LDU)^T = U^T D^T L^T = LDU = A$ because A is symmetric. Since $D^T = D$ (it is a diagonal matrix) we have that $L = U^T \equiv L^T = U$. So we rewrite $A = LDU = LDL^T$ where $L \in \mathbb{R}^{n\times n}$ is a unit lower-triangular matrix and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive diagonal entries (see part (B) for why D has positive diagonal entries).

(5) A more efficient algorithm for computing the factorization $A = LDL^T$.

Since computing U from LU factorization is not an efficient algorithm for symmetric positive definite matrices, we try to find an algorithm that exploits the symmetry of the matrix by working on either the upper or lower triangular submatrix. The algorithm will use the lower triangular part of the matrix to store the elements of L and leave the diagonals to be the elements of D.

Input: A, a symmetric positive definite matrix.

Starting from column 1 (j = 1), we want to set elements below the diagonal element (a_{11}) to be the multipliers as in LU factorization. That is, $l_{i1} = \frac{a_{i1}}{a_{11}}$. Then we want to compute the diagonal elements by subtracting this multiplier times the correct element in A as in LU factorization. This is where we exploit the symmetry of the matrix because we have altered $a_{i1} = l_{i1}$. So we use a_{1i} instead because it still contains the original element from A. Finally we repeat this for all columns $j = 1, ..., n$ to obtain the elements of L in the lower-triangular part of our matrix and the elements of D in the diagonal.

Output: L and D such that $A = LDL^T$.

(6) The Matlab program implementing the algorithm in part (5) is given below.

Part6.m

```
%%% Part (6)
% An algorithm for efficient LDL^T factorization for symmetric positive
% definite matrices

clear L D
format long e

% cd 'C:\Users\Christopher\Desktop\MAT 128B\Project 2';
A = input('Input the matrix: ');

if (size(A,1) ~= size(A,2))
  error('A is not symmetric')
end

n = size(A,1);
Noper = 0;

for j = 1:n
  % Sets the multiples
  A(j+1:n,j) = A(j+1:n,j)/A(j,j);
  Noper = Noper + 1;
  % Altering the entries on and below the diagonal as in LU factorization
  % Note that we are not computing anything in the upper-triangular matrix!
  for i = j+1:n
    % Can be slightly optimized here if we were doing something like LLT
    % instead of LDLT (save 1 multiplication per loop)
    A(i:n,i) = A(i:n,i) - A(i:n,j)*A(i,j)*A(j,j);
    Noper = Noper + 3;
  end
end
L = tril(A,-1);
% Sets the diagonal of L to be 1's
L(logical(eye(size(L)))) = 1;
D = diag(A);
D = diag(D);

fprintf('Done. The matrices are in L and D\n');
```

Christopher Wong
999234204
MAT 128B
Programming Project 2 **2/15/2017**

(7) Running the LDL^T program on

$$A = \begin{bmatrix} 26 & 12 & -42 & -20 \\ 12 & 36 & -18 & -18 \\ -42 & -18 & 81 & 33 \\ -20 & -18 & 33 & 19 \end{bmatrix}$$

we obtain:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4.615384615384616e-01 & 1 & 0 & 0 \\ -1.615384615384615e+00 & 4.545454545454553e-02 & 1 & 0 \\ -7.692307692307693e-01 & -2.878787878787879e-01 & 8.333333333333297e-02 & 1 \end{bmatrix}$$

$$D = \left[\begin{matrix} 2.600000000000000e+01 & 0 & 0 & 0 \\ 0 & 3.046153846153846e+01 & 0 & 0 \\ 0 & 0 & 1.309090909090908e+01 & 0 \\ 0 & 0 & 0 & 1.000000000000001e+00 \end{matrix}\right.$$

$N_{oper} = 22.$

Christopher Wong
999234204
MAT 128B
Programming Project 2 **2/15/2017**

(8) Running the LDL^T program on the 100×100 matrix A provided in A.mat we find

L	D
$l_{2,1} = -3.806285190567384e - 02$	$d_{1,1} = 1.386520904220077e + 01$
$l_{5,3} = -3.829937079554067e - 03$	$d_{7,7} = 1.687362175323990e + 01$
$l_{20,19} = 7.622625245899138e - 02$	$d_{10,10} = 5.758135998171008e + 00$
$l_{35,19} = 4.304249068140835e - 02$	$d_{17,17} = 6.052261534348436e + 00$
$l_{53,48} = -8.034550332972654e - 02$	$d_{20,20} = 1.272056815296482e + 01$
$l_{70,2} = -2.615456509230549e - 02$	$d_{35,35} = 1.049718371669377e + 01$
$l_{80,37} = -2.250770581899178e - 01$	$d_{41,41} = 5.168398366509415e + 00$
$l_{85,81} = 1.573872806026676e - 01$	$d_{66,66} = 2.398938164266977e + 00$
$l_{96,56} = 8.284241478139053e - 02$	$d_{83,83} = 1.385547930528885e + 00$
$l_{100,99} = 5.848276383604594e - 02$	$d_{100,100} = 3.712668203435592e - 02$

$N_{oper} = 14950.$