Direct Methods for Special Linear Systems

LDL Decomposition for Symmetric Matrices

If $\mathbf{A}$ is symmetric, then the LDM decomposition may be reduced to

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T.$$

Theorem 5. If $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{M}^T$ is the LDM decomposition of a nonsingular symmetric $\mathbf{A}$, then $\mathbf{L} = \mathbf{M}$.

Solving LDL:

- recall that in the previous LDM decomposition, the key is to find the unknown

  $$\mathbf{v} = \mathbf{D}\mathbf{M}^T\mathbf{e}_j$$

  by solving $\mathbf{A}_{1:j,j} = \mathbf{L}_{1:j,1:j}\mathbf{v}_{1:j}$ via forward substitution.
- Finding $\mathbf{v}$ is much easier and there is no need to run forward substitution.
  - (exploit the symmetry property) since $\mathbf{M} = \mathbf{L}$,

    $$v_i = d_i \ell_{ji}.$$

    All the elements, except for v_j, are known.
  - $a_{jj} = \mathbf{L}_{j,1:j}\mathbf{v}_{1:j} = \mathbf{L}_{j,1:j-1}\mathbf{v}_{1:j-1} + v_j = \mathbf{L}_{j,1:j-1}\mathbf{D}_{1:j-1,1:j-1}\mathbf{L}_{j,1:j-1}^T + v_j$

An LDL Decomposition Code

```
function [L,D]= my_ldl(A)
n= size(A,1);
L= eye(n); d= zeros(n,1); M= eye(n);
v= zeros(n,1);
for j=1:n,
     v(1:j)= for_subs(L(1:j,1:j),A(1:j,j));
     v(1:j-1)= L(j,1:j-1)'.*d(1:j-1); % replace for_subs.
     v(j)= A(j,j)- L(j,1:j-1)*v(1:j-1); % replace for_subs.
     d(j)= v(j);
     for i=1:j-1,
          M(j,i)= v(i)/d(i);
     end;
     L(j+1:n,j)= (A(j+1:n,j)-L(j+1:n,1:j-1)*v(1:j-1))/v(j);
end;
D= diag(d);
```

- complexity: $\mathcal{O}(n^3/3)$, half of LU or LDM
- LDL is used to solve symmetric linear systems

Cholesky Factorization for PD Matrices

- a matrix $\mathbf{A} \in \mathbb{S}^n$ is said to be positive semidefinite (PSD) if

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \quad \text{for all } \mathbf{x} \in \mathbb{R}^n;$$

  and positive definite (PD) if

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0, \quad \text{for all } \mathbf{x} \in \mathbb{R}^n \text{ with } \mathbf{x} \neq \mathbf{0}$$

Cholesky factorization: given a PD $\mathbf{A} \in \mathbb{S}^n$, factorize $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{G}\mathbf{G}^T,$$

where $\mathbf{G} \in \mathbb{R}^{n \times n}$ is lower triangular with positive diagonal elements and is called the Cholesky factor of $\mathbf{A}$.

- the factorization is also written as $\mathbf{A} = \mathbf{R}^T\mathbf{R}$ with upper triangular $\mathbf{R} \in \mathbb{R}^{n \times n}$
- we only discuss symmetric PD matrices here

Cholesky Factorization for PD Matrices

Theorem 6. If $\mathbf{A} \in \mathbb{S}^n$ is PD, then there exists a unique lower triangular $\mathbf{G} \in \mathbb{R}^{n\times n}$ with positive diagonal elements such that $\mathbf{A} = \mathbf{G}\mathbf{G}^T$.

- idea: if $\mathbf{A}$ is symmetric and PD, then its LDL decomposition

  $$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T$$

  has $d_i > 0$ for all $i = 1, \ldots, n$ (as an exercise, verify this). Putting $\mathbf{G} = \mathbf{L}\mathbf{D}^{\frac{1}{2}}$ where $\mathbf{D}^{\frac{1}{2}} = \mathrm{Diag}(d_1^{\frac{1}{2}}, \ldots, d_n^{\frac{1}{2}})$ yields the Cholesky factorization.

Solving Cholesky factorization:

- (exploit the symmetry) the key is to find the unknown

  $$\mathbf{v} = \mathbf{G}^T\mathbf{e}_j \quad \text{or} \quad v_i = g_{ji}.$$

  All the elements, except for v_j, are known.
- (exploit the positive-definiteness property)

  $$\begin{aligned} a_{jj} =& \mathbf{G}_{j,1:j}\mathbf{v}_{1:j} = \mathbf{G}_{j,1:j-1}\mathbf{v}_{1:j-1} + g_{jj}v_j = \mathbf{G}_{j,1:j-1}\mathbf{G}_{j,1:j-1}^T + g_{jj}^2 \\ =& \mathbf{v}_{1:j-1}^T\mathbf{v}_{1:j-1} + (v_j)^2 \end{aligned}$$

A Cholesky Factorization Code

```
function [G]= my_Cholesky(A)
n= size(A,1);
G= zeros(n,n);
v= zeros(n,1);
for j=1:n,
     v(1:j-1)= G(j,1:j-1);
     v(j)= sqrt(A(j,j)- v(1:j-1)'*v(1:j-1));
     G(j,j)= v(j);
     G(j+1:n,j)= (A(j+1:n,j)-G(j+1:n,1:j-1)*v(1:j-1))/v(j);
end;
```

- computing procedure is similar to LDL
- can be computed in $\mathcal{O}(n^3/3)$, no pivoting required, numerically very stable
- Cholesky decomposition is used to solve PD linear systems

Pivoted Cholesky Factorization

Pivoted Cholesky factorization: given a PSD $\mathbf{A} \in \mathbb{S}^n$, factorize $\mathbf{A}$ as

$$\mathbf{P}^T\mathbf{A}\mathbf{P} = \mathbf{G}\mathbf{G}^T,$$

where $\mathbf{P}$ is a permutation matrix, and

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix} \in \mathbb{R}^{n \times r}$$

with leading submatrix $\mathbf{G}_1 \in \mathbb{R}^{r \times r}$ being lower triangular with positive diagonal.

- r_{ii} can be chosen to satisfy $r_{11} \geq r_{22} \geq \cdots \geq r_{rr} > 0$
- $\mathrm{rank}(\mathbf{A}) = \mathrm{rank}(\mathbf{G}) = \mathrm{rank}(\mathbf{G}_1) = r$

LU Decomposition for Band Matrices

For a banded matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$,

- lower bandwidth p if $a_{ij} = 0$ whenever $i > j + p$
- upper bandwidth q if $a_{ij} = 0$ whenever $j > i + q$

Theorem 7. Suppose $\mathbf{A} \in \mathbb{R}^{n \times n}$ has an LU factorization $\mathbf{A} = \mathbf{LU}$. If $\mathbf{A}$ has lower bandwidth p and upper bandwidth q, then $\mathbf{L}$ has lower bandwidth p and $\mathbf{U}$ has upper bandwidth q.

Proof: cf. Theorem 4.3.1 in **[Golub-van-Loan'13]** for details

- $\mathbf{L}$ inheritates the lower bandwidth of $\mathbf{A}$
- $\mathbf{U}$ inheritates the upper bandwidth of $\mathbf{A}$

Banded LU factorization with partial pivoting: the upper bandwidth of $\mathbf{U}$ is $p + q$
cf. Theorem 4.3.2 in **[Golub-van-Loan'13]** for details

Iterative Methods for Linear Systems

Iterative Methods for Linear Systems

- such iterative methods are a.k.a. indirect methods
- solving linear systems via LU requires $\mathcal{O}(n^3)$
- $\mathcal{O}(n^3)$ is too much for large-scale linear systems
- the motivation behind iterative methods is to seek less expensive ways to find an (approximate) linear system solution
- note: see also the ideas of handling large-scale LS problems forthcoming in LS Topic, which is relevant to the context here

The Key Insight of Iterative Methods

- assume $a_{ii} \neq 0$ for all i
- observe

$$\begin{aligned} \mathbf{b} = \mathbf{A}\mathbf{x} \quad & \Longleftrightarrow \quad b_i = a_{ii} x_i + \sum_{j \neq i} a_{ij} x_j, \quad i = 1, \ldots, n \\ & \Longleftrightarrow \quad x_i = \left(b_i - \sum_{j \neq i} a_{ij} x_j \right) / a_{ii}, \quad i = 1, \ldots, n \qquad (\dagger) \end{aligned}$$

- idea: find an $\mathbf{x}$ that fulfils the equations in $(\dagger)$

Jacobi Iterations

input: a starting point $\mathbf{x}^{(0)}$
for $k = 0, 1, 2, \ldots$
  par_for $i = 1, 2, \ldots, n$
    $x_i^{(k+1)} = \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right) / a_{ii}$
  end
end

- complexity per iteration: $\mathcal{O}(n^2)$ for dense $\mathbf{A}$, $\mathcal{O}(\mathrm{nnz}(\mathbf{A}))$ for sparse $\mathbf{A}$
- the Jacobi update step can be computed in a parallel or distributed fashion
  - same idea appeared in distributed power control in 2G or 3G wireless networks
- a natural idea, heuristic at first glance
- does the Jacobi iterations converge to the linear system solution?
  - it does not, in general
  - it does if the diagonal elements a_{ii}'s are "dominant" compared to the off-diagonal elements; see Theorem 11.2.2 in **[Golub-van-Loan'13]** for details

Gauss-Seidel (G-S) Iterations

input: a starting point $\mathbf{x}^{(0)}$
for $k = 0, 1, 2, \ldots$
  for $i = 1, 2, \ldots, n$

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) / a_{ii}$$

  end
end

- use the most recently available $\mathbf{x}$ to perform update
- sequential, cannot be computed in a distributed or parallel manner
- coordinatewise minimization, a special case of coordinate descent (CD) method
- guaranteed to converge to the linear system solution if
  - $\mathbf{A}$ has diagonally dominant characteristics (similar to the Jacobi iterations)
  - $\mathbf{A}$ is symmetric PD; see Theorem 11.2.3 in **[Golub-van-Loan'13]**

Minimization Methods

- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a symmetric and positive definite matrix. In this case, solving the linear system $\mathbf{Ax} = \mathbf{b}$ is equivalent to

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c,$$

  for an arbitrary scalar constant $c \in \mathbb{R}$.

- many minimization methods: gradient descent, steepest descent, conjugate gradient descent, preconditioned conjugate gradients, ADMM, etc.

Other Topics on Linear Systems

Consistent and Inconsistent Systems

In algebra, a linear or nonlinear system of equations is called consistent if it possesses at least one solution. If there are no solutions, the system is called inconsistent.

Problem: Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$,

$$\begin{aligned} \text{find} \quad & \mathbf{x} \in \mathbb{R}^n \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

- the linear system is consistent if and only if

$$\mathbf{b} \in \mathcal{R}(\mathbf{A})$$

- under-determined when $m < n$: either infinitely many solutions or no solutions
- well-determined or exactly determined when $m = n$: unique, infinitely many, or no solutions
- over-determined when $m > n$: unique, infinitely many, or no solutions

Solution of Linear Systems

Let $\mathbf{A}$ be m-by-n and $\mathrm{rank}(\mathbf{A}) = r < n$. Then there is an $n - r$ dimensional set of vectors $\mathbf{x}$ that satisfy $\mathbf{Ax} = \mathbf{b}$.

Proof. Let $\mathbf{Az} = \mathbf{0}$. Then if $\mathbf{x}$ satisfies $\mathbf{Ax} = \mathbf{b}$, so does $\mathbf{x} + \mathbf{z}$.

Underdetermined Systems

Problem: If $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, $\mathrm{rank}(\mathbf{A}) = m$, and $\mathbf{b} \in \mathbb{R}^m$, find $\mathbf{x} \in \mathbb{R}^n$ s.t.

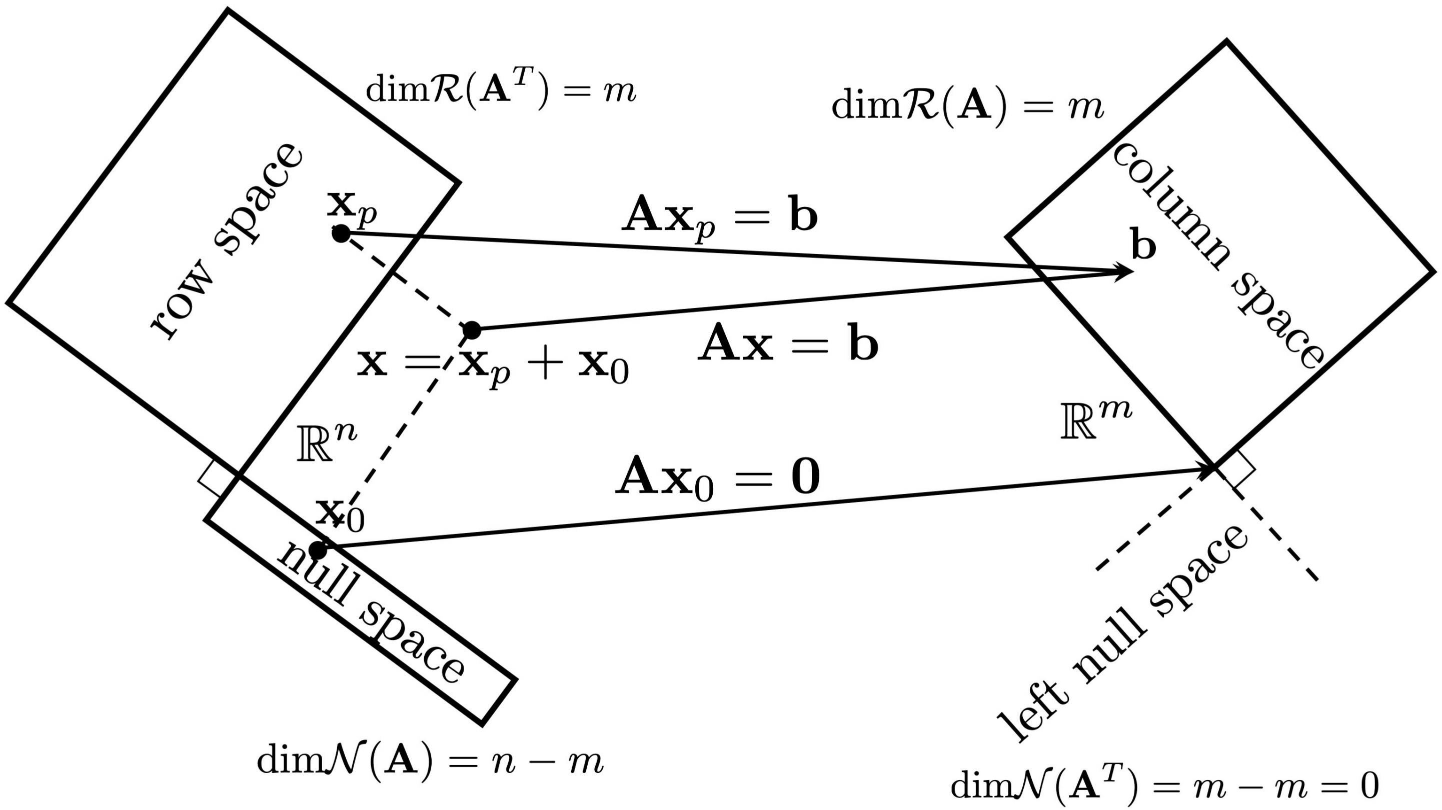
$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

- it is always true that $\mathbf{b} \in \mathcal{R}(\mathbf{A})$
- an underdetermined linear system has infinite number of solutions given by

$$\mathbf{x} = \mathbf{x}_p + \mathbf{x}_0 = \mathbf{x}_p + \mathbf{F}\mathbf{v} \quad \text{with} \quad \mathbf{v} \in \mathbb{R}^{n-m},$$

  where $\mathbf{x}_p \in \mathcal{R}(\mathbf{A}^T)$ is (any) particular solution and special solutions $\mathbf{x}_0 \in \mathcal{N}(\mathbf{A})$ with columns of $\mathbf{F} \in \mathbb{R}^{n \times (n-m)}$ spans $\mathcal{N}(\mathbf{A})$.
- several numerical methods for computing $\mathbf{F}$ (rectangular LU decomposition, QR factorization (cf. QR Topic), ...)
- solution to smallest ℓ_2 norm: $\mathbf{x}_0 = \mathbf{0}$, i.e., $\mathbf{v} = \mathbf{0}$, cf. SVD Topic
- solution to smallest ℓ_0 "norm": can we find a sparsest solution $\mathbf{x}$? cf. Compressive Sensing Topic

Underdetermined Systems



Note: there is a counterpart mapping from the right to left corresponding to $\mathbf{A}^T$.

Solving Underdetermined Systems via Rectangular LU

A rectangular LU decomposition of $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{L}\big[\mathbf{U}_1 \ \mathbf{U}_2\big]$$

where $\mathbf{L} \in \mathbb{R}^{m\times m}$ is unit lower triangular, $\mathbf{U}_1 \in \mathbb{R}^{m\times m}$ is nonsingular and uppertriangular, and $\mathbf{U}_2 \in \mathbb{R}^{m\times(n-m)}$.

- note

  $$\mathbf{A}\mathbf{x} = \mathbf{L}\big[\mathbf{U}_1 \ \mathbf{U}_2\big]\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} = \mathbf{L}(\mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_2) = \mathbf{b}$$

  which can be solved by first solving $\mathbf{L}\mathbf{z} = \mathbf{b}$ and then solving $\mathbf{U}_1\mathbf{x}_1 = \mathbf{z} - \mathbf{U}_2\mathbf{x}_2$ given a specific $\mathbf{x}_2 \in \mathbb{R}^{n-m}$, we have $\mathbf{x}_1 = \mathbf{U}_1^{-1}\mathbf{L}^{-1}\mathbf{b} - \mathbf{U}_1^{-1}\mathbf{U}_2\mathbf{x}_2$. Then,

  $$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} = \begin{bmatrix}\mathbf{U}_1^{-1}\mathbf{L}^{-1}\mathbf{b} - \mathbf{U}_1^{-1}\mathbf{U}_2\mathbf{x}_2\\ \mathbf{x}_2\end{bmatrix} = \begin{bmatrix}\mathbf{U}_1^{-1}\mathbf{L}^{-1}\mathbf{b}\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}-\mathbf{U}_1^{-1}\mathbf{U}_2\\ \mathbf{I}\end{bmatrix}\mathbf{x}_2$$

- So, one solution is to set $\mathbf{x}_p = \begin{bmatrix}\mathbf{U}_1^{-1}\mathbf{L}^{-1}\mathbf{b}\\ \mathbf{0}\end{bmatrix}$, $\mathbf{F} = \begin{bmatrix}-\mathbf{U}_1^{-1}\mathbf{U}_2\\ \mathbf{I}\end{bmatrix}$, and $\mathbf{v} = \mathbf{x}_2$.

What if $\mathbf{b} \notin \mathcal{R}(\mathbf{A})$?

When $\mathbf{b} \notin \mathcal{R}(\mathbf{A})$, we can find an $\mathbf{x}$ such that $\mathbf{A}\mathbf{x}$ is closer to $\mathbf{b}$ via

$$\min_{\mathbf{x} \in \mathbb{R}^n} \ \rho(\mathbf{b} - \mathbf{A}\mathbf{x}) \tag{LS}$$

where $\rho : \mathbb{R}^m \to \mathbb{R}$ denotes a distance function.

- ℓ_2 norm: least squares (LS) problem (cf. Least Squares Topic)
- ℓ_1 norm: least absolute deviations (LAD)
- divergence measures
- other loss functions...

Sensitivity Analysis of Linear Systems

- Scenario:
  - let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be nonsingular, and $\mathbf{b} \in \mathbb{R}^n$. Let $\mathbf{x}$ be the solution to

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

  - consider a perturbed version of the above system: $\hat{\mathbf{A}} = \mathbf{A} + \Delta\mathbf{A}, \hat{\mathbf{b}} = \mathbf{b} + \Delta\mathbf{b}$, where $\Delta\mathbf{A}$ and $\Delta\mathbf{b}$ are errors. Let $\hat{\mathbf{x}}$ be a solution to the perturbed system

$$\hat{\mathbf{A}}\hat{\mathbf{x}} = \hat{\mathbf{b}}.$$

- Problem: analyze how the solution error $\|\hat{\mathbf{x}} - \mathbf{x}\|_2$ scales with $\Delta\mathbf{A}$ and $\Delta\mathbf{b}$
- remark: $\Delta\mathbf{A}$ and $\Delta\mathbf{b}$ may be floating point errors, measurement errors, etc
- forthcoming in SVD Topic.

References

[Golub-van-Loan'13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th edition, JHU Press, 2013.

[Horn-Johnson'12] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, 2nd edition, Cambridge University Press, 2012.