Linear Least Squares and Numerical Methods

Guo Yuanxin

May 22 2019

Contents

- Different Cases of LLS
- Numerical Methods
  - Normal equations – Cholesky
  - Orthogonal – QR
  - Orthogonal – SVD
- Comparison
  - Numerical stability
  - Time complexity
- Methods for Sparse Solutions – MP & OMP
- Examples

Preliminary: Vector Derivatives

If $f(\mathbf{x})$ is a function of $x_1, \ldots, x_N$, $\dfrac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial f(\mathbf{x})}{\partial x_1} \\ \dfrac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \dfrac{\partial f(\mathbf{x})}{\partial x_N} \end{bmatrix}$.

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{b} = \mathbf{b} \qquad \frac{\partial}{\partial \mathbf{x}} \mathbf{b}^T \mathbf{x} = \mathbf{b}.$$

Suppose that $\mathbf{A}$ is a symmetric real matrix, $\mathbf{A}^T = \mathbf{A}$. $\dfrac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}$.

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{y} - \mathbf{x})^T \mathbf{A} (\mathbf{y} - \mathbf{x}) = 2\mathbf{A}(\mathbf{x} - \mathbf{y}), \quad \frac{\partial}{\partial \mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = 2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}).$$

Case I: Overdetermined System

- Overdetermined: No solution, usually $\mathbf{A}$ tall with linearly independent columns.

- Minimize: $r(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$

- $\frac{\partial}{\partial \mathbf{x}} r(\mathbf{x}) = 0 \quad \Rightarrow \quad \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}$

- $\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y}$ or $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{y}$

Case II: Underdetermined System

- Underdetermined: Many solutions $\mathbf{B}$ B wide with independent rows .

- Find the solution with smallest norm. i.e., $\min_{\mathbf{x}} \|\mathbf{x}\|_2^2 \quad s.t. \quad \mathbf{y} = \mathbf{B}\mathbf{x}$

- Lagrange multipliers: $\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = \|\mathbf{x}\|_2^2 - \boldsymbol{\mu}^T(\mathbf{y} - \mathbf{B}\mathbf{x})$

- $\mathbf{x} = \mathbf{B}^T \left(\mathbf{B}\mathbf{B}^T\right)^{-1} \mathbf{y}$ or $\mathbf{x} = \mathbf{B}^{\dagger}\mathbf{y}$

Case III: Regularization

- Minimize: $c_1\|\mathbf{y}-\mathbf{M}\mathbf{x}\|_2^2+c_2\|\mathbf{x}\|_2^2$
- Solution depend on c_1/c_2 , but not on them independently.

- $\mathcal{J}(\mathbf{x})=\|\mathbf{y}-\mathbf{M}\mathbf{x}\|_2^2+\lambda\|\mathbf{x}\|_2^2$

- $\frac{\partial}{\partial\mathbf{x}}\mathcal{J}(\mathbf{x})=0 \quad \Rightarrow \quad \left(\mathbf{M}^T\mathbf{M}+\lambda\mathbf{I}\right)\mathbf{x}=\mathbf{M}^T\mathbf{y} \quad \Rightarrow \quad \mathbf{x}=\left(\mathbf{M}^T\mathbf{M}+\lambda\mathbf{I}\right)^{-1}\mathbf{M}^T\mathbf{y}$

- Former two: Full rank; Regularization: Handles rank deficiency.

Normal Equations – Definiteness of a Matrix

Definition 3.1 (Positive definite)**.** A symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is *positive definite* if

$$\mathbf{x}^T\mathbf{M}\mathbf{x} > 0, \quad \forall \mathbf{x} \in \mathbb{R}^n, \ \mathbf{x} \neq 0$$

Proposition 3.1. *If* $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m > n$, *has rank* n, *then* $\mathbf{A}^T\mathbf{A}$ *is positive definite.*

$$\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|^2 > 0, \quad \forall x \in \mathbb{R}^n, \ x \neq 0$$

Normal Equations – Cholesky Factorization

Definition 3.2 (Schur complement)**.** Partition an $n \times n$ matrix $\mathbf{A}$ as

$$\mathbf{A} = \begin{bmatrix} A_{11} & {A_{2:n,1}}^T \\ A_{2:n,1} & A_{2:n,2:n} \end{bmatrix}$$

The *Schur complement* of $\mathbf{A}_{11}$ is defined as the $n-1 \times n-1$ matrix $\mathbf{S}$

$$\mathbf{S} = A_{2:n,2:n} - \frac{1}{A_{11}} A_{2:n,1} {A_{2:n,1}}^T$$

Theorem 3.2 (Positive definiteness of $\mathbf{S}$)**.** *If* $\mathbf{A}$ *is positive definite, then* $\mathbf{S}$ *is also positive definite.*

Normal Equations – Cholesky Factorization

Theorem 3.3 (Cholesky Factorization). *Every positive definite matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *can be factorized as*

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

where $\mathbf{L}$ *is a lower triangular matrix.*

Written explicitly, the factorization looks like:

$$\begin{aligned}
\begin{bmatrix} A_{11} & {A_{2:n,1}}^T \\ A_{2:n,1} & A_{2:n,2:n} \end{bmatrix} &= \begin{bmatrix} L_{11} & 0 \\ L_{2:n,1} & L_{2:n,2:n} \end{bmatrix} \begin{bmatrix} L_{11} & {L_{2:n,1}}^T \\ 0 & {L_{2:n,2:n}}^T \end{bmatrix} \\
&= \begin{bmatrix} {L_{11}}^2 & L_{11}{L_{2:n,1}}^T \\ L_{11}L_{2:n,1} & L_{2:n,1}{L_{2:n,1}}^T + L_{2:n,2:n}{L_{2:n,2:n}}^T \end{bmatrix}
\end{aligned}$$

Normal Equations – Cholesky Factorization

Algorithm 3.1 (Cholesky Decomposition)**.**

1. $L_{11} = \sqrt{A_{11}}$
2. $L_{2:n,1} = A_{2:n,1}/A_{11}$
3. $L_{2:n,2:n}{L_{2:n,2:n}}^T = A_{2:n,2:n} - {L_{2:n,1}}^T L_{2:n,1}$
4. Repeat 1-3 for $L_{2:n,2:n}{L_{2:n,2:n}}^T$

- Time Complexity: $\frac{1}{3}n^3$ flops

Normal Equations: Algorith,

Algorithm 3.2 (Normal Equations Method).

1.Calculate $\mathbf{C} = \mathbf{A}^T\mathbf{A}$.

2.Cholesky Factorization: $\mathbf{C} = \mathbf{L}\mathbf{L}^T$.

3.Calculate $\mathbf{d} = \mathbf{A}^T\mathbf{b}$.

4.Solving $\mathbf{L}\mathbf{z} = \mathbf{d}$ by forward substitution.

5.Solving $\mathbf{L}^T\mathbf{x} = \mathbf{z}$ by back substitution.

- Total flops: $\sim mn^2 + \frac{1}{3}n^3 \; flops$

Orthogonal – QR: Triangular LLS

$$\begin{bmatrix} R \\ O \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

where R is an $n \times n$ upper triangular matrix partition, O is the zero $(m-n) \times n$ matrix partition. The m-vector $\begin{bmatrix} b_1 & b_2 \end{bmatrix}^T$ is partitioned in a similar manner.

The linear least square problem becomes minimizing:

$$\|r\|_2^2 = \|b_1 - R\mathbf{x}\|_2^2 + \|b_2\|_2^2$$

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|r\|_2^2 = \|b_2\|_2^2,$$

$$\mathbf{x} = R^{-1} b_1$$

Orthogonal – QR: A Theorem

Theorem 3.4. *Given the matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and the vector* $\mathbf{b} \in \mathbb{R}^m$. *Assume*

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} R \\ O \end{bmatrix} \tag{43}$$

is the QR factorization of $\mathbf{A}$, *where* $\mathbf{Q} \in \mathbb{R}^{m \times m}$, $R \in \mathbb{R}^{n \times n}$. *The solution set to the linear least squares problem*

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 \tag{44}$$

is identical to the solution set of

$$R\mathbf{x} = \left(\mathbf{Q}^T \mathbf{b}\right)_{1:n} \tag{45}$$

Orthogonal – QR: Householder Transformations

Algorithm 3.3 (QR Factorization via Householder Reflectors).

for $k = 1$ **to** n

$$v = A_{k:m,k}$$
$$u_k = v - \|v\| e_1$$
$$u_k = u_k / \|u_k\|$$
$$A_{k:m,k:n} = A_{k:m,k:n} - 2u_k({u_k}^T A_{k:m,k:n})$$

- Total flops: $\sum_{k=1}^{n} 4(m-k+1)(n-k+1) \sim 2mn^2 - \frac{2}{3}n^3 \ flops$

Orthogonal – QR: Algorithm

Note that we don't explicitly calculate $\mathbf{Q}$ or $\mathbf{Q}^T$ because it takes too many flops. When we want to calculate $\mathbf{Qx}$ or $\mathbf{Q}^T\mathbf{b}$, we simply do the following:

Algorithm 3.4 (Calculation of $\mathbf{Q}^T\mathbf{b}$).

for $k = 1$ **to** n
$$b_{k:m} = b_{k:m} - 2u_k({u_k}^T b_{k:m})$$

Algorithm 3.5 (Calculation of $\mathbf{Qx}$).

for $k = n$ **downto** 1
$$x_{k:m} = x_{k:m} - 2u_k({u_k}^T x_{k:m})$$

Orthogonal – QR: Algorithm

Algorithm 3.6 (Orthogonal Method Using QR).

1. Compute the upper triangular $\mathbf{R}$ such that $\mathbf{A} = \mathbf{QR}$.
2. Compute $\hat{b}_1 = (\mathbf{Q}^T\mathbf{b})_{1:n}$.
3. Solve $\mathbf{R}_{1:n,1:n}\mathbf{x} = \hat{b}_1$.

- **Total flops:** $\sim 2mn^2 - \frac{2}{3}n^3 \ flops$

Comparison – Time Complexity

1. Normal equations $\sim\ mn^2 + n^3/3$ flops
2. QR factorization $\sim\ 2mn^2 - 2n^3/3$ flops
3. SVD decomposition $\sim\ 2mn^2 + 11n^3$ flops

- $m \approx n$ Normal ~ QR << SVD
- $m \gg n$, Normal << QR ~ SVD

Sparse Solution

- Aim: Find an *approximate* solution with at most k nonzero entries.

- Finding either the most sparse or the closest: Too time consuming

- Greedy algorithm: Matching Pursuit

Matching Pursuit

- Greedy & Iterative

- Each iteration:
  - Find the column that has the greatest effect on the residual
  - Calculate the coefficient
  - Update residual (eliminate the most correlated column’s effect)

Matching Pursuit

- Preparation: Normalize the column vectors to be unit norm. This is because we need to compare the correlation between a vector and the column vectors. Normalizing makes it more convenient to compare.
- Initialization: Set residual $r_1 = b$, iteration counter $i = 1$, index set $\Lambda_0 = \emptyset$.
- The main loop: **for** $i = 1$ **to** k
  - Find the column vector whose inner product with residual r_i has the greatest absolute value, that is:

$$\gamma_i = \arg\max_j |\langle r_i, a_j \rangle|, \ j = 1, \cdots, n. \tag{115}$$

  - Update index set $\Lambda_i = \Lambda_{i-1} \cup \{\gamma_i\}$
  - Calculate coefficient $c_i = \langle r_i, a_{\gamma_i} \rangle$.
  - Update residual $r_{i+1} = r_i - c_i a_{\gamma_i}$.
  - Update counter $i = i + 1$.
- Return: List of coefficients $\{c_i\}$ and index list Λ_i.

The sparse *approximate* solution of this LLS problem is:

$$\sum_{\gamma_i \in \Lambda_k} c_i e_{\gamma_i}, \tag{116}$$

A Variant – Orthogonal Matching Pursuit

- Variation:
  - MP: Calculate a coefficient each iteration
  - OMP: Update all coefficients that has been calculated in each iteration.

- Advantage:
  - Find the best coefficients for a given index list.
  - Converges quicker in iterations

- Drawback:
  - Requires much more flops.
  - Converges slower in flops

Examples

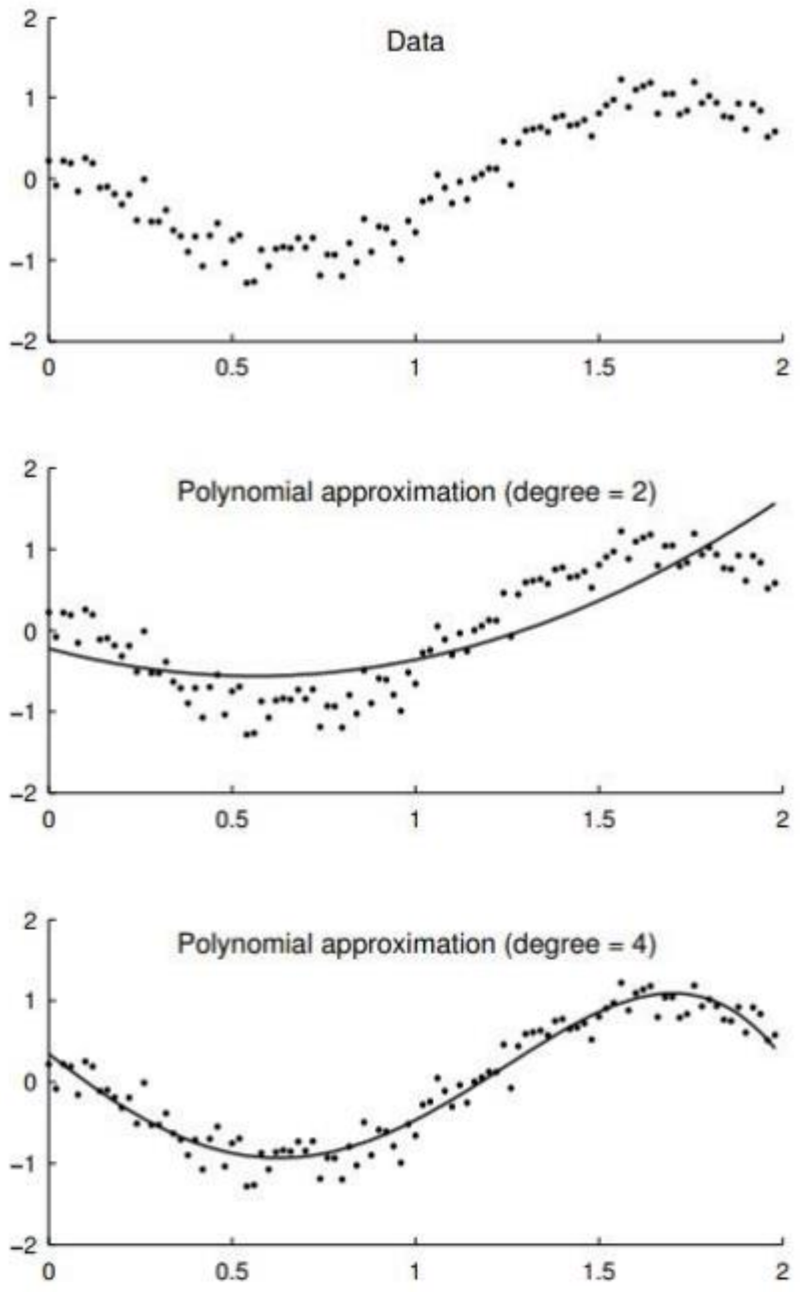
- Polynomial approximation

Suppose that the polynomial is of degree 2, *i.e.*,

$$p(x) = a_0 + a_1 x + a_2 x^2. \tag{118}$$

The minimization problem can be viewed as solving the overdetermined LLS problem given by:

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}. \tag{119}$$



Examples

- Smoothing
- Noisy Signal: $\mathbf{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}$
- Find: $\min_{\hat{\mathbf{y}}} \|\hat{\mathbf{y}} - \mathbf{y}\|^2 + \lambda \|\mathbf{D}\hat{\mathbf{y}}\|^2,$

The function to be minimized consists of two terms. Minimizing the first term ensures $\hat{\mathbf{y}}$ is similar to $\mathbf{y}$. The extreme case is when $\lambda = 0$, whose result is $\hat{\mathbf{y}} = \mathbf{y}$. Minimizing the second term ensures smoothness. Choosing appropriate λ balances both requirements.

Measuring the smoothness of a signal is usually by computing its discrete counterpart of second-order derivative, second-order difference. Define the matrix $\mathbf{D}$ as

$$\mathbf{D} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix}. \tag{123}$$

Then $\mathbf{Dy}$ is the second-order difference of $\mathbf{y}$.