Linear Least Squares and Numerical Methods

CUHK(SZ)

Guo Yuanxin

May 21 2019

Contents

1 Introduction

In this report we will mainly discuss linear least squares (LLS) problems. The discussion will be divided into a few parts. The first part is aimed to introduce the different cases of LLS. In the second part we will use different methods to solve LLS problems, namely Normal Equation, QR, and SVD. We will then compare their numerical stability and time complexity. In the third part we will introduce methods for sparse solutions to LLS problems. In the last part, we will briefly introduce some of the applications of LLS problems.

2 Different Cases of LLS

2.1 Overdetermined systems

Consider the overdetermined linear system:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{1}$$

Overdetermined means there is no solution to the system. In many cases, it means that $\mathbf{A}$ is a "tall" matrix with linearly independent columns. Here we want to find a $\mathbf{x}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$ is "closest" to $\mathbf{y}$. In other words, we want to minimize

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

Expand the terms,

$$\begin{aligned} r(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 &= (\mathbf{y} - \mathbf{A}\mathbf{x})^T(\mathbf{y} - \mathbf{A}\mathbf{x}) && (3)\\ &= \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{A}^T\mathbf{y} + \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} && (4)\\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} && (5) \end{aligned}$$

Taking the derivative of $r(\mathbf{x})$,

$$\frac{\partial}{\partial \mathbf{x}} r(\mathbf{x}) = -2\mathbf{A}^T\mathbf{y} + 2\mathbf{A}^T\mathbf{A}\mathbf{x} \tag{6}$$

Let the derivative equal to 0, we get

$$\frac{\partial}{\partial \mathbf{x}} r(\mathbf{x}) = 0 \quad \Rightarrow \quad \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y} \tag{7}$$

The formula we have just derived is always known as the Normal Equation. Note that we can reach the same result by a geometric approach.
Assume further that our $\mathbf{A}^T\mathbf{A}$ is of full rank, that is, invertible. We can rewrite formula (7) as:

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y} \tag{8}$$

$\mathbf{x}$ is the least squares solution.

2.2 Underdetermined systems

Consider the underdetermined linear system:

$$\mathbf{y} = \mathbf{B}\mathbf{x} \tag{9}$$

Underdetermined means there are many solutions to the problem. This frequently happens when $\mathbf{B}$ is a wide matrix with linearly independent rows. In these problems, we often seek the solution with the smallest norm. The optimization problem becomes:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_2^2 \;\; s.t. \;\; \mathbf{y} = \mathbf{B}\mathbf{x} \tag{10}$$

The method of Lagrange multipliers is often used in constrained optimization. We here define the Lagrangian $\mathcal{L}(\mathbf{x}, \boldsymbol{\mu})$ as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = \|\mathbf{x}\|_2^2 - \boldsymbol{\mu}^T(\mathbf{y} - \mathbf{B}\mathbf{x}) \tag{11}$$

Take the derivatives of the Lagrangian,

$$\frac{\partial}{\partial \mathbf{x}}\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = 2\mathbf{x} - \mathbf{B}^T\boldsymbol{\mu} \tag{12}$$

$$\frac{\partial}{\partial \boldsymbol{\mu}}\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{y} - \mathbf{B}\mathbf{x} \tag{13}$$

By letting them equal to zero,

$$\mathbf{x} = \frac{1}{2}\mathbf{B}^T\boldsymbol{\mu} \tag{14}$$

$$\mathbf{y} = \mathbf{B}\mathbf{x} \tag{15}$$

Plugging (14) into (15), we get,

$$\mathbf{y} = \frac{1}{2}\mathbf{B}\mathbf{B}^T\boldsymbol{\mu} \tag{16}$$

If we assume further that $\mathbf{B}\mathbf{B}^T$ is invertible, then we can write the minimal norm solution $\mathbf{x}$ to the underdetermined system as:

$$\boldsymbol{\mu} = 2\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{y} \tag{17}$$

$$\mathbf{x} = \mathbf{B}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{y} \tag{18}$$

Remark. The least squares solutions in the first two cases can be denoted as $\mathbf{x} = \mathbf{A}^\dagger\mathbf{y}$ or $\mathbf{x} = \mathbf{B}^\dagger\mathbf{y}$, where $\mathbf{A}^\dagger$, $\mathbf{B}^\dagger$ are the *Moore-Penrose pseudoinverses.*

2.3 Regularization

In the previous parts, we tried to minimize $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ and $\|\mathbf{x}\|_2^2$ in an overdetermined system and in an underdetermined system respectively. For a general

linear system $\mathbf{y} = \mathbf{M}\mathbf{x}$, another approach is to minimize the weighted sum: $c_1\|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2^2 + c_2\|\mathbf{x}\|_2^2$. Note that the solution only depend on the ratio c_1/c_2, but not on c_1 and c_2 individually.
A common approach for solving this type of problems is minimizing the objective function $\mathcal{J}(\mathbf{x})$:

$$\mathcal{J}(\mathbf{x}) = \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 \tag{19}$$

Taking the derivative,

$$\frac{\partial}{\partial \mathbf{x}}\mathcal{J}(\mathbf{x}) = 2\mathbf{M}^T(\mathbf{M}\mathbf{x} - \mathbf{y}) + 2\lambda\mathbf{x} \tag{20}$$

Setting the derivative to zero,

$$\frac{\partial}{\partial \mathbf{x}}\mathcal{J}(\mathbf{x}) = 0 \quad \Rightarrow \quad \mathbf{M}^T\mathbf{M}\mathbf{x} + \lambda\mathbf{x} = \mathbf{M}^T\mathbf{y} \tag{21}$$

$$\Rightarrow \quad \left(\mathbf{M}^T\mathbf{M} + \lambda\mathbf{I}\right)\mathbf{x} = \mathbf{M}^T\mathbf{y} \tag{22}$$

Note that $\lambda > 0$ and $\mathbf{M}^T\mathbf{M}$ is positive semi-definite. So that $\left(\mathbf{M}^T\mathbf{M} + \lambda\mathbf{I}\right)$ is positive definite, which implies invertibility.
Thus we can write the solution to this minimization problem as:

$$\mathbf{x} = \left(\mathbf{M}^T\mathbf{M} + \lambda\mathbf{I}\right)^{-1}\mathbf{M}^T\mathbf{y} \tag{23}$$

This approach is known as diagonal loading, since a constant λ is added to the diagonal. The approach effectively evades the problem of rank deficiency of $\mathbf{M}$, *i.e.*, $\mathbf{M}^T\mathbf{M}$ being singular.

3 Numerical Methods of LLS

In this part we will introduce three different methods of solving linear least squares problem. For the sake of simplicity, we will illustrate the methods in terms of solving LLS in an overdetermined system.

3.1 Normal Equations Method

We have stated that $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}$ is referred to as the "Normal Equation". Now we aim to find a numerically efficient way to solve an equation like this.

3.1.1 Definiteness of a Matrix

Before we introduce the numerical solution to the problem, we first introduce some important concept.

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

$$\mathbf{x}^T\mathbf{M}\mathbf{x} > 0, \quad \forall \mathbf{x} \in \mathbb{R}^n,\ \mathbf{x} \neq 0 \tag{24}$$

We now propose that the operator $\mathbf{A}^T\mathbf{A}$ in the left hand side of the normal equation is positive definite if $\mathbf{A} \in \mathbb{R}^{m\times n},\ m > n$ is of full rank. In other words, $rank(\mathbf{A}) = n$.

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.*

To see this, we just need to see that:

$$\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|^2 > 0, \quad \forall x \in \mathbb{R}^n,\ x \neq 0 \tag{25}$$

Since $\mathbf{A}$ has full rank.
In the following part, we will focus only on the case when $\mathbf{A}$ has full rank.

3.1.2 Cholesky Factorization

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

$$\mathbf{A} = \begin{bmatrix} A_{11} & {A_{2:n,1}}^T \\ A_{2:n,1} & A_{2:n,2:n} \end{bmatrix} \tag{26}$$

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

$$\mathbf{S} = A_{2:n,2:n} - \frac{1}{A_{11}} A_{2:n,1}{A_{2:n,1}}^T \tag{27}$$

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

Proof. To see this, we consider any $\mathbf{x} \in \mathbb{R}^{n-1} \neq 0$ and define $\mathbf{y} = -\left({A_{2:n,1}}^T\mathbf{x}\right)/A_{11}$. Then,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T \mathbf{A} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}^T \begin{bmatrix} A_{11} & {A_{2:n,1}}^T \\ A_{2:n,1} & A_{2:n,2:n} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \tag{28}$$

$$= \mathbf{y}^T A_{11}\mathbf{y} + 2\mathbf{y}^T {A_{2:n,1}}^T\mathbf{x} + \mathbf{x}^T A_{2:n,2:n}\mathbf{x} \tag{29}$$

$$= -\left(\frac{1}{A_{11}} {A_{2:n,1}}^T\mathbf{x}\right)^T {A_{2:n,1}}^T\mathbf{x} + \mathbf{x}^T A_{2:n,2:n}\mathbf{x} \tag{30}$$

$$= \mathbf{x}^T\mathbf{S}\mathbf{x} > 0 \tag{31}$$

■

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

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T \tag{32}$$

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

Written explicitly, the factorization looks like:

$$\begin{bmatrix} A_{11} & {A_{2:n,1}}^T \\ A_{2:n,1} & A_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{2:n,1} & L_{2:n,2:n} \end{bmatrix} \begin{bmatrix} L_{11} & {L_{2:n,1}}^T \\ 0 & {L_{2:n,2:n}}^T \end{bmatrix} \tag{33}$$

$$= \begin{bmatrix} {L_{11}}^2 & L_{11}{L_{2:n,1}}^T \\ L_{11}L_{2:n,1} & L_{2:n,1}{L_{2:n,1}}^T + L_{2:n,2:n}{L_{2:n,2:n}}^T \end{bmatrix} \tag{34}$$

We calculate $\mathbf{L}$, the Cholesky factor by an recursive procedure.

Algorithm 3.1 (Cholesky Decomposition)**.**

1. $L_{11} = \sqrt{A_{11}}$
2. $L_{2:n,1} = A_{2:n,1}/\sqrt{A_{11}}$
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 $L_{2:n,2:n}{L_{2:n,2:n}}^T$

The validity of the recursive process is guaranteed since we have shown the Schur complement of A_{11}, $\mathbf{S}$ is also positive definite if $\mathbf{A}$ is positive definite.
The time complexity of this factorization is $\sim n^3/3$. We can calculate this by calculating flops of each recursive layer. In the first layer, steps 1-3 requires

$$1 + (n-1) + 2 \times \frac{n(n+1)}{2} \ \sim \ n^2 \tag{35}$$

Note that the last term comes from symmetry of $\mathbf{A}$.
So the total complexity is roughly:

$$\sum_{k=1}^{n} k^2 \sim \frac{1}{3} n^3 flops \tag{36}$$

3.1.3 Normal Equations Method

As we have introduced the numerical way to solve normal equations, we can now implement it to solve LLS problems. Generally, solving a LLS problem with $\mathbf{A}$ having full rank can be divided into the following steps.

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.

Step 1 takes $\sim mn^2$ flops, since $\mathbf{C}$ is symmetric. Step 3 requires $\sim 2mn$ flops, and Step 4 and 5 both requires $\sim n^2$ flops. Asymptotically, the total flops

required to complete this algorithm is:

$$\sim mn^2 + \frac{1}{3}n^3 \ flops \tag{37}$$

3.2 Orthogonal Methods - The QR Factorization

QR Factorization is an alternative method to solve linear least squares problem. We continue to illustrate the overdetermined, full-rank case first.

3.2.1 Triangular Least Squares Problem

Consider a triangular overdetermined system

$$\begin{bmatrix} R \\ O \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{38}$$

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:

$$\|r\|_2^2 = \|b_1 - R\mathbf{x}\|_2^2 + \|b_2\|_2^2 \tag{39}$$

Remember that we assumed that $\begin{bmatrix} R & O \end{bmatrix}^T$ has full rank n, that is to say, R has full rank n. This implies the invertibility of the square, upper-triangular matrix R. Since both terms in $\|r\|_2^2$ are non-negative, the minimum of $\|r\|_2^2$ is obtained when the first term $\|b_1 - R\mathbf{x}\|_2^2 = 0$. Thus,

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|r\|_2^2 = \|b_2\|_2^2, \tag{40}$$

$$\mathbf{x} = R^{-1} b_1 \tag{41}$$

This previous argument gives rise to the Orthogonal Method of solving linear least squares problem.

3.2.2 The QR Factorization in Linear Least Squares

Recall the full QR decomposition of a $m \times n$ matrix $\mathbf{A}$,

$$\mathbf{A} = \mathbf{QR}, \quad \mathbf{Q} \in \mathbb{R}^{m \times m}, \mathbf{R} \in \mathbb{R}^{m \times n} \tag{42}$$

where $\mathbf{Q}$ is an orthogonal matrix, and $\mathbf{R}$ is "upper triangular". We now state a straightforward 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

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

Proof. Using the properties of orthogonal matrices, we have:

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \|\mathbf{Q}^T(\mathbf{b} - \mathbf{A}\mathbf{x})\|_2^2 \qquad \text{(Orthogonality of Q}^{\mathrm{T}}\text{)} \tag{46}$$

$$= \left\|\mathbf{Q}^T\mathbf{b} - \begin{bmatrix} R \\ O \end{bmatrix}\mathbf{x}\right\|_2^2 \qquad \text{(QR factorization)} \tag{47}$$

Partition $\mathbf{Q}^T\mathbf{b}$ as:

$$\mathbf{Q}^T\mathbf{b} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \end{bmatrix} \tag{48}$$

where $\hat{b}_1$ is of length n, $\hat{b}_2$ is of length $(m-n)$. Then we have:

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \left\|\begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \end{bmatrix} - \begin{bmatrix} R \\ O \end{bmatrix}\mathbf{x}\right\|_2^2 \tag{49}$$

$$= \left\|\begin{bmatrix} \hat{b}_1 - R\mathbf{x} \\ \hat{b}_2 \end{bmatrix}\right\|_2^2 \tag{50}$$

From our previous argument, we have already seen that the minimum of the residual $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$ is obtained when $\hat{b}_1 - R\mathbf{x} = 0$, which is identically the solutions to $R\mathbf{x} = \left(\mathbf{Q}^T\mathbf{b}\right)_{1:n}$ ■

3.2.3 Implementing the Orthogonal Method Using QR

We have seen multiple ways to implement QR factorization, including Gram-Schmidt Orthogonalization, Jacobi Rotation, and Householder Reflection. Due to time and space limitations, we do not analyze them comparatively in this text. Among the methods, Householder Reflection is numerically efficient and stable. As we have seen it before, we only introduce it briefly.

Definition 3.3 (Householder Reflector)**.** A Householder transformation of a vector $\mathbf{x}$ is its reflection with respect to a plane (or hyperplane) π through the origin with a unit normal vector $\mathbf{v}$. It is represented by

$$\mathbf{x}' = \mathbf{x} - 2\mathbf{v}\mathbf{v}^T\mathbf{x}. \tag{51}$$

$\mathbf{P}_\mathbf{v} = (\mathbf{I} - 2\mathbf{v}\mathbf{v}^T)$ is the *Householder reflector* corresponding to the transformation.

$\mathbf{P}$ is orthogonal and symmetric. This is easily verified. Interested readers can justify on your own. For a general normal vector $\mathbf{u}$ of a plane, the Householder reflector can be written as:

$$\mathbf{P} = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} \tag{52}$$

If we let $\mathbf{u} = \mathbf{x} - \|\mathbf{x}\|\,\mathbf{e}_1$, where $\mathbf{x}$ is any vector, the effect of applying $\mathbf{P}$ to the vector $\mathbf{x}$ is reflecting the vector to the direction of $\mathbf{e}_1$ while preserving norm. We verify it:

$$\begin{aligned}
\|\mathbf{u}\|^2 &= (\mathbf{x} - \|\mathbf{x}\|\,\mathbf{e}_1)^T (\mathbf{x} - \|\mathbf{x}\|\,\mathbf{e}_1) && (53)\\
&= \mathbf{x}^T\mathbf{x} - 2\|\mathbf{x}\|\mathbf{x}^T\mathbf{e}_1 + \|\mathbf{x}\|^2\mathbf{e}_1^T\mathbf{e}_1 && (54)\\
&= 2\left(\|\mathbf{x}\|^2 - \|\mathbf{x}\|x_1\right) && (55)\\
\mathbf{P}\mathbf{x} &= \left(\mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2}\right)\mathbf{x} && (56)\\
&= \left(\mathbf{I} - 2\frac{\mathbf{u}\,(\mathbf{x} - \|\mathbf{x}\|\,\mathbf{e}_1)^T}{\|\mathbf{u}\|^2}\right)\mathbf{x} && (57)\\
&= \mathbf{x} - 2\frac{\mathbf{u}\left(\|\mathbf{x}\|^2 - \|\mathbf{x}\|x_1\right)}{2\left(\|\mathbf{x}\|^2 - \|\mathbf{x}\|x_1\right)} && (58)\\
&= \mathbf{x} - \mathbf{u} && (59)\\
&= \|\mathbf{x}\|\mathbf{e}_1 && (60)
\end{aligned}$$

This actually gives us some sense in constructing the matrix $\mathbf{Q}^T$. We first construct a series of orthogonal matrices $\mathbf{Q}_k$, where each of them could be written as

$$\begin{bmatrix} \mathbf{I} & O \\ O & \mathbf{P} \end{bmatrix}, \tag{61}$$

where $\mathbf{I}$ is the $(k-1)\times(k-1)$ identity matrix and $\mathbf{P}$ is the $(m-k+1)\times(m-k+1)$ Householder reflector we have just constructed. Applying them one by one to $\mathbf{A}$ and setting particular $\mathbf{u}$'s, we obtain an upper triangular $\mathbf{R}$. Thus

$$\mathbf{Q}^T = \mathbf{Q}_n \cdots \mathbf{Q}_2\mathbf{Q}_1. \tag{62}$$

By symmetry of $\mathbf{P}$ and hence symmetry of $\mathbf{Q}_k$,

$$\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2 \cdots \mathbf{Q}_n. \tag{63}$$

The entries of $\mathbf{R}$ can be obtained by the following iterative algorithm:

Algorithm 3.3 (QR Factorization via Householder Reflectors)**.**

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

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

Most work is done in the last step of the loop, which takes $4(m-k+1)(n-k+1)$ flops. Thus asymtotically, the algorithm costs:

$$\sum_{k=1}^{n} 4(m-k+1)(n-k+1) \sim 2mn^2 - \frac{2}{3}n^3 \ flops \tag{64}$$

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
$x_{k:m} = x_{k:m} - 2u_k({u_k}^T x_{k:m})$

Therefore, the whole algorithm for the Orthogonal Method using QR is:

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 $\mathbf{R}_{1:n,1:n}\mathbf{x} = \hat{b}_1$.

Since the time complexity of Step 2 is $\mathcal{O}(mn)$ and the time complexity of Step 3 is $\mathcal{O}(n^2)$, the total time complexity of this algorithm is asymptotically same with finishing the QR Factorization, that is,

$$\sim 2mn^2 - \frac{2}{3}n^3 \ flops \tag{65}$$

3.2.4 Rank Deficient Case and Minimum Norm Solution

Recall that we assumed $\mathbf{A} \in \mathbb{R}^{m\times n}$ has full rank n in the previous parts. Now we consider the case when rank($\mathbf{A}$)= $r < n$. The QR factorization would now look like:

$$\mathbf{A} = \mathbf{Q}\begin{bmatrix} R_1 & R_2 \\ O & O \end{bmatrix}, \tag{66}$$

where $\mathbf{Q} \in \mathbb{R}^{m\times m}$, $R_1 \in \mathbb{R}^{r\times r}$ and is nonsingular, upper triangular, $R_2 \in \mathbb{R}^{r\times(n-r)}$. The least squares problem is now converted to minimizing:

$$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2^2 = \|\mathbf{Q}^T(\mathbf{b}-\mathbf{A}\mathbf{x})\|_2^2 \tag{67}$$

$$= \|\mathbf{Q}^T\mathbf{b} - \begin{bmatrix} R_1 & R_2 \\ O & O \end{bmatrix}\mathbf{x}\|_2^2 \tag{68}$$

Using similar strategy, we partition $\mathbf{Q}^T\mathbf{b}$ as:

$$\mathbf{Q}^T\mathbf{b} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix}, \tag{69}$$

where $\hat{b}_1$ is of length r, $\hat{b}_2$ is of length $n-r$, $\hat{b}_3$ is of length $m-n$.
We then partition $\mathbf{x}$ as $[\ {x_1}^T\ \ {x_2}^T\ ]^T$, where x_1 is of length r, x_2 is of length $n-r$.
This gives us

$$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2^2 = \left\| \begin{bmatrix} \hat{b}_1 - R_1x_1 - R_2x_2 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} \right\|_2^2 \tag{70}$$

$$= \|\hat{b}_1 - R_1x_1 - R_2x_2\|_2^2 + \|\hat{b}_2\|_2^2 + \|\hat{b}_3\|_2^2 \tag{71}$$

The solution to this minimization problem is

$$x_1 = {R_1}^{-1}\left(\hat{b}_1 - R_2x_2\right), \tag{72}$$

for any $x_2 \in \mathbb{R}^{n-r}$. There are infinitely many solution, which is in accord to our assumption that $\mathbf{A}$ has rank $r < n$.
Among all solutions, we focus on the one with the smallest 2-norm, which is called the *minimum norm solution*. Finding this solution means solving

$$\min_{x_2\in\mathbb{R}^{n-r}} \left\| \begin{bmatrix} {R_1}^{-1}\left(\hat{b}_1 - R_2x_2\right) \\ x_2 \end{bmatrix} \right\|_2^2 = \tag{73}$$

$$\min_{x_2\in\mathbb{R}^{n-r}} \left\| \begin{bmatrix} {R_1}^{-1}\hat{b}_1 \\ 0 \end{bmatrix} - \begin{bmatrix} {R_1}^{-1}R_2 \\ I \end{bmatrix} x_2 \right\|_2^2 \tag{74}$$

This, again, is a linear least square problem and one with full rank. We can solve this using previous methods.

3.3 Orthogonal Methods - SVD Decomposition

The gist of using QR factorization is to convert a general linear system to a triangular system. Now we introduce Singular Value Decomposition (SVD), which converts a linear system to a diagonal system.

3.3.1 Singular Value Decomposition

We state the following theorem without proof.

Theorem 3.5 (Singular Value Decomposition)**.** *For any matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, there exists orthogonal matrices* $\mathbf{U} \in \mathbb{R}^{m\times m}$*,* $\mathbf{V} \in \mathbb{R}^{n\times n}$ *and a 'diagonal' matrix* $\Sigma \in \mathbb{R}^{m\times n}$ *with diagonal entries*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_{min\{m,n\}} = 0 \tag{75}$$

such that $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$*. The diagonal entries* $\{\sigma_i\}$ *are called singular values.*

Remark. Using the orthogonality, we can rewrite the decomposition as:

$$\mathbf{AV} = \mathbf{U}\Sigma \tag{76}$$

Also, we can see the rank of $\mathbf{A}$ equals the number of nonzero diagonal entries of Σ, or, rank($\mathbf{A}$)$= r$.

Now we aim to solve LLS with SVD.

3.3.2 Finding Minimum Norm Solution Using SVD

Consider the general LLS (overdetermined or underdetermined, not necessarily full-rank):

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \tag{77}$$

Let $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ be the SVD of $\mathbf{A}$.
Using the orthogonality of $\mathbf{U}$ and $\mathbf{V}$, we get

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|_2^2 = \min_{\mathbf{x}\in\mathbb{R}^n} \left\|\mathbf{U}^T\left(\mathbf{AVV}^T\mathbf{x} - \mathbf{b}\right)\right\|_2^2 \tag{78}$$

$$= \min_{\mathbf{x}\in\mathbb{R}^n} \left\|\Sigma\mathbf{V}^T\mathbf{x} - \mathbf{U}^T\mathbf{b}\right\|_2^2 \tag{79}$$

By setting $\mathbf{V}^T\mathbf{x} = \mathbf{z}$,

$$\|\mathbf{Ax} - \mathbf{b}\|_2^2 = \sum_{i=1}^{r}\left(\sigma_i z_i - {u_i}^T b\right)^2 + \sum_{j=r+1}^{m}\left({u_j}^T b\right)^2 \tag{80}$$

Thus, a general solution is given by:

$$z_i = \begin{cases} \dfrac{{u_i}^T b}{\sigma_i}, & i = 1, \cdots, r. \\ arbitrary, & i = r+1, \cdots, n. \end{cases} \tag{81}$$

$$\mathbf{x} = \sum_{i=1}^{n} z_i v_i \tag{82}$$

Since $\mathbf{V}$ is orthogonal and that $\mathbf{z} = \mathbf{V}^T\mathbf{x}$, $\|\mathbf{z}\|_2^2 = \|\mathbf{x}\|_2^2$. This implies finding the minimum norm solution, $\mathbf{x}_\dagger$, is equivalent to minimizing the 2-norm of $\mathbf{z}$. This is simply completed by choosing $z_i = 0,\ i = r+1, \cdots, n$. Therefore, the minimum norm solution of this LLS is given by:

$$\mathbf{x}_\dagger = \sum_{i=1}^{r} \frac{{u_i}^T b}{\sigma_i} v_i \tag{83}$$

The computation of SVD is an interesting subject in itself, so we are not discussing it here. The number of flops required to compute the SVD is

$$\sim 2mn^2 + 11n^3 \ \ flops. \tag{84}$$

4 Comparison of Methods

We will briefly compare the time complexity and numerical stability in this part. Before doing this, we shall first introduce some concepts for error analysis.

4.1 Matrix Norms

We are familiar with the *Frobenius norm* of a $m \times n$ matrix $\mathbf{A}$, which is numerically equal to the 2-norm of the mn-vector constructed by stacking up all the column vectors a_i. However, there is a more useful definition of matrix norm in practice.

Definition 4.1 (Induced matrix norm)**.** Given vector norms $\|\cdot\|_{(n)}$ and $\|\cdot\|_{(m)}$ on the domain and range of $\mathbf{A} \in \mathbb{R}^{m\times n}$ respectively.The *induced matrix norm* $\|A\|_{(m,n)}$ is the smallest number C for which the following inequality holds for all $\mathbf{x} \in \mathbb{R}^n$:

$$\|\mathbf{Ax}\|_{(m)} \leq C\|\mathbf{x}\|_{(n)}. \tag{85}$$

In other words,

$$\|\mathbf{A}\|_{(m,n)} = \sup_{\mathbf{x}\in\mathbb{R}^n, \mathbf{x}\neq 0} \frac{\|\mathbf{Ax}\|_{(m)}}{\|\mathbf{x}\|_{(n)}} \tag{86}$$

$$= \sup_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|_{(n)}=1} \|\mathbf{Ax}\|_{(m)} \tag{87}$$

We say that $\|\cdot\|_{(m,n)}$ is the matrix norm induced by $\|\cdot\|_{(m)}$ and $\|\cdot\|_{(n)}$

Remark. The definition is also valid for $\mathbf{B} \in \mathbb{C}^{m\times n}$, but we are only interested in real matrices in this text.

Remark. When $m = n = p$, we simply say the "p-norm" of a matrix, denoted as $\|\cdot\|_p$.

The most commonly used p-norm of a matrix is its 2-norm. A more succinct definition of the matrix 2-norm is given by:

$$\|\mathbf{A}\|_2 = \sigma_1, \tag{88}$$

where σ_1 is the largest singular value of $\mathbf{A}$. This is intuitive by the motivation of SVD decomposition. Now we introduce two important inequalities concerning matrix norms.

Theorem 4.1. *For any $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times k}$, the following inequalities hold:*

$$\|\mathbf{Ax}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{x}\|_p \tag{89}$$

$$\|\mathbf{AB}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{B}\|_p \tag{90}$$

Proof. The first inequality is directly from the definition of induced matrix norms. For the second inequality, note that for any $\mathbf{x} \in \mathbb{R}^k$

$$\|\mathbf{ABx}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{Bx}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{B}\|_p \|\mathbf{x}\|_p. \tag{91}$$

Divide by $\|\mathbf{x}\|_p$ on both sides and taking the supremum, we have the required inequality (90). ■

4.2 Error Analysis

In real life problems, it is usual for the data we obtained to have perturbations. We are often concerned about whether a numerical solution is sensitive to perturbations. We consider a solvable linear system

$$\mathbf{Ax} = \mathbf{b} \tag{92}$$

where $\mathbf{A}$ is invertible.
We first consider perturbation only in $\mathbf{b}$. The perturbed system now become

$$\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b} + \Delta\mathbf{b}. \tag{93}$$

If we let $\tilde{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{x}$, where $\Delta\mathbf{x}$ stands for the absolute error, we get

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}. \tag{94}$$

Subtracting (92) by (94), we have

$$\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b} \qquad \text{or} \qquad \Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b} \tag{95}$$

Now we want to find a upper bound of the relative error expressed in terms of perturbation. Recall the inequality (89) we derived in the last part. Take norms of (95):

$$\|\Delta\mathbf{x}\| = \|\mathbf{A}^{-1}\mathbf{b}\| \leq \|\mathbf{A}^{-1}\|\|\Delta\mathbf{b}\|. \tag{96}$$

Also, we have

$$\|\mathbf{b}\| = \|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\| \quad \Rightarrow \quad \frac{1}{\|\mathbf{x}\|} \leq \|\mathbf{A}\|\frac{1}{\|\mathbf{b}\|} \tag{97}$$

Combining the two formulae, we have find an upper bound for the relative error $\|\Delta\mathbf{x}\|/\|\mathbf{x}\|$:

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} \tag{98}$$

Definition 4.2 (Condition number)**.** The (p-)*condition number* $\kappa_p(\mathbf{A})$ of a nonsingular matrix $\mathbf{A}$ is defined by

$$\kappa_p(\mathbf{A}) = \|\mathbf{A}\|_p\|\mathbf{A}^{-1}\|_p. \tag{99}$$

If

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{100}$$

and

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}, \tag{101}$$

then the relative error obeys

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} = \kappa_p(\mathbf{A})\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}. \tag{102}$$

The definition does not apply for rectangular matrices, since the inverse is not defined. Recall the *Moore-Penrose pseudoinverse*:

$$\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T, \text{ (Tall)}, \tag{103}$$

$$\mathbf{A}^\dagger = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}, \text{ (Wide)}. \tag{104}$$

For a general full-rank matrix $\mathbf{A}'$, the p-condition number of $\mathbf{A}'$ is defined as

$$\kappa_p(\mathbf{A}') = \|\mathbf{A}'\|_p\|(\mathbf{A}')^\dagger\|_p. \tag{105}$$

It is easy using (90) to check that $\kappa_p(\mathbf{A}') \geq 1$ for any nonsingular matrix $\mathbf{A}'$. In particular, $\kappa_p(\mathbf{I}) = 1$. If $\kappa_p(\mathbf{A}')$ is small, we say that the linear system is *well-conditioned*. Otherwise, we say that the linear system is *ill-conditioned*. The more ill-conditioned a linear system is, the more sensitive it is to perturbations.

4.3 Comparing the Numerical Methods of LLS

4.3.1 Numerical Stability

In this part all matrix norms are 2-norms, which is not to be specified later. Considering the overdetermined, full-rank LLS problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2, \ \mathbf{A} \in \mathbb{R}^{m\times n}, \ \mathbf{b} \in \mathbb{R}^m. \tag{106}$$

Assume $\mathbf{A}$ has full QR factorization:

$$\mathbf{A} = \mathbf{QR},\ \mathbf{Q} \in \mathbb{R}^{m\times m},\ \mathbf{R} \in \mathbb{R}^{m\times n} \tag{107}$$

and SVD decomposition:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T,\ \mathbf{U} \in \mathbb{R}^{m\times m}, \Sigma \in \mathbb{R}^{m\times n},\ \mathbf{V} \in \mathbb{R}^{n\times n}. \tag{108}$$

We have shown in previous text that the three methods convert the LLS problem to solving the following linear systems:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b} \qquad \text{(Normal Equations)} \tag{109}$$

$$\mathbf{R}_{1:n,1:n}\mathbf{x} = (\mathbf{Q}^T\mathbf{b})_{1:n} \qquad \text{(QR factorization)} \tag{110}$$

$$\Sigma_{1:n,1:n}\mathbf{V}^T\mathbf{x} = (\mathbf{U}^T\mathbf{b})_{1:n} \qquad \text{(SVD decomposition)} \tag{111}$$

Note that the entries of $\mathbf{R}_{n+1:m,1:n}$ and $\Sigma_{n+1:m,1:n}$ are all zero. Thus $\|\mathbf{R}_{1:n,1:n}\| = \|\mathbf{R}\|$, $\|\Sigma_{1:n,1:n}\| = \|\Sigma\|$.
Since $\mathbf{Q}, \mathbf{U}, \mathbf{V}$ are all orthogonal matrices, they are all norm-preserving. So we conclude:

$$\|\mathbf{A}\| = \|\mathbf{R}_{1:n,1:n}\| = \|\Sigma_{1:n,1:n}\mathbf{V}^T\|. \tag{112}$$

Using similar rationale, we can show that the condition number κ for the matrices in the two orthogonal methods are the same as $\kappa(\mathbf{A})$. However, in the normal equations method,

$$\kappa(\mathbf{A}^T\mathbf{A}) = \|\mathbf{A}^T\mathbf{A}\|\|\mathbf{A}^{-1}\mathbf{A}^{-T}\| \sim [\kappa(\mathbf{A})]^2, \tag{113}$$

which implies that the Normal Equations method is very unstable when the problem is ill-conditioned.

4.3.2 Time Complexity

The number of flops required to solve a LLS is listed below:

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

When $m \approx n$, the normal equations method and the QR factorization methods takes approximately the same amount of operations, while the cost of SVD is too much compared to the first two methods.
When $m \gg n$, if the problem is well-conditioned and efficiency is the major concern, the normal equations method is the most desirable. The QR factorization and the SVD decomposition method cost approximately the same amount of work, as the first term $2mn^2$ dominates the operation count.

5 Sparse Solution of Underdetermined LLS

In this section, we consider solving the underdetermined linear system:

$$\mathbf{b} = \mathbf{A}\mathbf{x}, \tag{114}$$

where $A \in \mathbb{R}^{m\times n},\ m < n,\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{b} \in \mathbb{R}^m$.
There are infinitely many solutions to this system, but we are not concerned with finding an accurate solution in this section. Instead, we are looking for an approximate solution that has at most k nonzero entries, or alternatively, a *k-sparse solution*. This is a common problem encountered in signal processing. Here we introduce an algorithm to resolve this problem.

5.1 Matching Pursuit

Idealistically, we want to find the most sparse solution such that $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|$ is less than a certain threshold. However, the amount of computation is enormous, which makes it impractical. Thus, Matching Pursuit (MP) was proposed. It is a greedy algorithm, which requires far less computation and returns an approximate solution. The algorithm works as the following:

- 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:

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

where e_{γ_i} is the standard unit vector with its ${\gamma_i}^{th}$ entry equal to one while other entries are all zero.

5.2 A Variant - Orthogonal Matching Pursuit

The advantages and drawbacks of Matching Pursuit are apparent: it is efficient, but the results it produces is not always satisfactory in accuracy. A popular extension of Matching Pursuit: Orthogonal Matching Pursuit (OMP) can give better results, but requires more computation.
The difference in MP and OMP lies in that MP computes one more coefficient in each iteration, while OMP updates *all* coefficients computed. We will now see how OMP works:

- Preparation: Normalize the column vectors.
- Initialization: Set residual $r_1 = b$, iteration counter $i = 1$, index set $\Lambda_0 = \emptyset$.
- The main loop: **for** $i = 1$ **to** k
  - $\gamma_i = \arg\max_j |\langle r_i, a_j \rangle|,\ j = 1, \cdots, n.$
  - Update index set $\Lambda_i = \Lambda_{i-1} \cup \{\gamma_i\}$
  - Calculate coefficient vector $\mathbf{v}_i = \arg\min_{\mathbf{v}} \|\mathbf{b} - \mathbf{A}_{\Lambda_i}\|$.
  - Update residual $r_{i+1} = r_i - \mathbf{A}_{\Lambda_i}\mathbf{v}_i$.
  - Update counter $i = i + 1$.
- Return: List of coefficients $\{c_i\}$ and index list Λ_i.

The improvement in OMP lies in that it guarantees to find the 'best' coefficients given an index list Λ, while MP does not.
A rough analysis of these two algorithm suggests that the time complexity of them are both $\mathcal{O}(mnk)$. This is established upon $k \ll m < n$, so that the first step of the loop consumes the most work. However, empirical data have shown that when setting the same threshold of the residuals, OMP needs much more flops than MP, despite the much less iterations it requires.

6 Some Examples of LLS

6.1 Low Order Polynomial Approximation

An important example of linear least squares is fitting a low-order polynomial to data. Suppose the N-point data is of the form (x_i, y_i) for $1 \leq i \leq N$. Our

goal is to find a polynomial $p(x)$ that approximates the data. This is realized by minimizing the energy of the residual:

$$E = \sum_{i=1}^{N} \left(y_i - p(x_i)\right)^2 . \tag{117}$$

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:

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

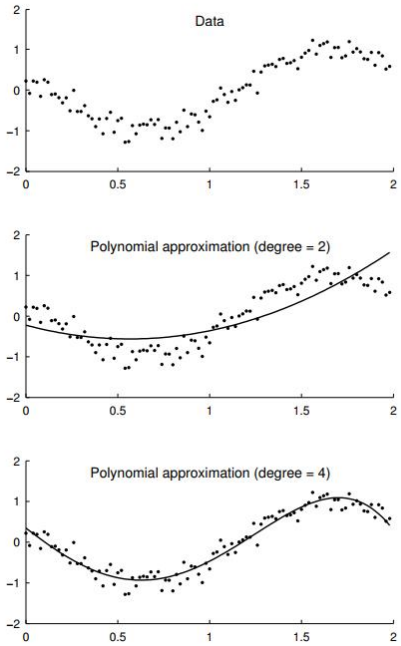

Figure 1: Least square polynomial approximation.

6.2 Linear Prediction

One approach to predict the future values of a time series is based on linear prediction. For example,

$$y_n \approx a_1 y_{n-1} + a_2 y_{n-2} + a_3 y_{n-3}. \tag{120}$$

If past data $y(n)$ for $0 \le n \le N-1$ is available, we can convert the problem of seeking an third order linear predictor to a overdetermined LLS problem, given by

$$\begin{bmatrix} y(3) \\ y(4) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} y(2) & y(1) & y(0) \\ y(3) & y(2) & y(1) \\ \vdots & \vdots & \vdots \\ y(N-2) & y(N-3) & y(N-4) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \tag{121}$$

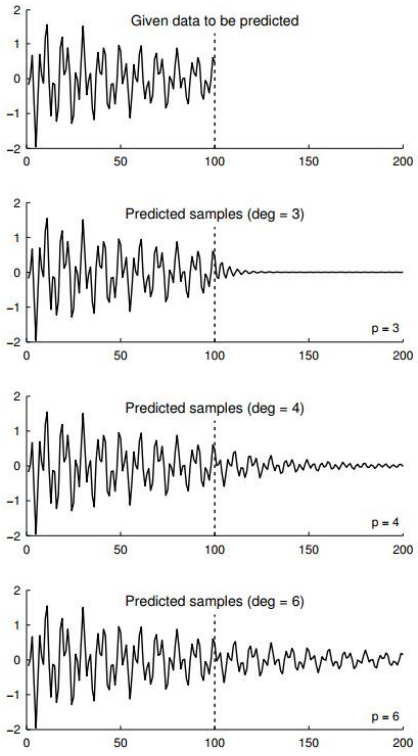

Figure 2: Least squares linear prediction.

6.3 Smoothing

In signal processing we always want to smooth a noisy signal. Consider the noisy signal

$$\mathbf{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \tag{122}$$

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 $\mathbf{y}$.

The problem of smoothing is actually of LLS problem with regularization. We want to find a smooth signal $\hat{\mathbf{y}}$, which is the solution to the following problem:

$$\min_{\hat{\mathbf{y}}} \|\hat{\mathbf{y}} - \mathbf{y}\|^2 + \lambda \|\mathbf{D}\hat{\mathbf{y}}\|^2, \tag{124}$$

where $\lambda > 0$ is a parameter to be specified.

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.

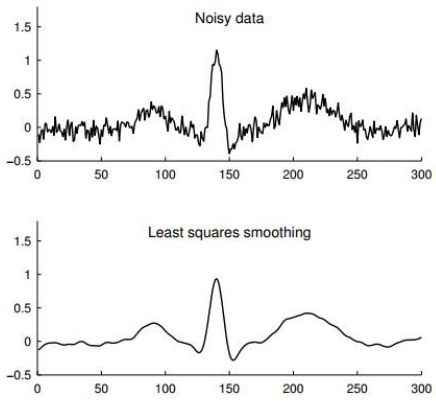

Figure 3: Least squares smoothing.