

# 1 Linear Models

## 1.1 Types of errors

Errors come from: imprecise data, mistakes in the model, computational precision,.. We know two types of errors:

- **Absolute** error = approximate value - correct value

$$\Delta x = \bar{x} - x$$

- **Relative** error =  $\frac{\text{absolute err}}{\text{correct value}}$

$$\delta x = \frac{\Delta x}{x}$$

**1.2** Mathematical model is **linear**, when the function F is a linear function of the parameters:

$$F(x, a_1, \dots, a_p) = a_1 \phi_1(x) + \dots + a_p \phi_p(x)$$

where  $\phi_1, \dots, \phi_p$  are functions of a specific type.

**1.3 Least squares method** Given points

$$\{(x_1, y_1), \dots, (x_m, y_m)\}, x_i \in R^n, y_i \in R$$

the task is to find a function  $F(x, a_1, \dots, a_p)$  that is good fit for the data. The values of the parameters  $a_1, \dots, a_p$  should be chosen so that the equations

$$y_i = F(x, a_1, \dots, a_p), i = 1, \dots, m$$

are satisfied or, if this is not possible, that the error is as small as possible.

We use **Least squares method** to determine that the sum of squared errors is as small as possible.

$$\sum_{i=1}^m (F(x_i, a_1, \dots, a_p) - y_i)^2$$

## 1.4 Systems of linear equations

A system of linear equations in the matrix form is given by  $A\vec{x} = \vec{b}$ , where:

- A is the matrix of coefficients of order  $m \times n$  where  $m$  is the number of equations and  $n$  is the number of unknowns,
- $\vec{x}$  is the vector of unknowns and
- $\vec{b}$  is the right side vector

$$\begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_p(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_p(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & \phi_p(x_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

## 1.5 Existence of solutions in linear equations

Let  $A = [\vec{a}_1, \dots, \vec{a}_n]$ , where  $\vec{a}_i$  are vector representing the columns of A. For any vector

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \text{ the product } A\vec{x} \text{ is a linear combination } A\vec{x} = \sum_i x_i a_i.$$

The system is **solvable** iff the vector  $\vec{b}$  can be expressed as a linear combination of the columns of A, that is it is in the column space of A,  $\vec{b} \in C(A)$ .

By adding  $\vec{b}$  to the columns of A we obtain the extended matrix of the system:

$$[A|\vec{b}] = [\vec{a}_1, \dots, \vec{a}_n | \vec{b}]$$

The system  $A\vec{x} = \vec{b}$  is solvable iff the rank of A equals the rank of the extended matrix  $[A|\vec{b}]$ , i.e.:

$$\text{rank} A = \text{rank}[A|\vec{b}] =: r$$

The solution is unique if the rank of the two matrices equals num of unknowns ( $r = n$ ).

## 1.6 Properties of squared matrices

Let  $A \in R^{n \times n}$  be a square matrix. The following conditions are equivalent and characterize when a matrix A is **invertible** or **non-singular**:

- The matrix A has an inverse
- $\text{rank } A = n$
- $\det(A) \neq 0$
- The null space  $N(A) = \{\vec{x} : A\vec{x} = 0\}$  is trivial
- All eigenvalues of A are nonzero
- For each  $\vec{b}$  the system of equations  $A\vec{x} = \vec{b}$  has precisely one solution

**1.7 Generalized inverse** of a matrix  $A \in R^{n \times m}$  is a matrix  $G \in R^{m \times n}$  such that

$$AGA = A$$

Let  $G$  be a generalized inverse of A. Multiplying  $AGA = A$  with  $A^{-1}$  from the left and the right side we obtain:

$$\text{LHS: } A^{-1}GAA^{-1} = IGI = G$$

$$\text{RHS: } A^{-1}AA^{-1} = IA^{-1} = A^{-1}$$

where I is the identity matrix. The equality LHS=RHS implies that  $G = A^{-1}$ .

Every matrix  $A \in R^{n \times m}$  has a generalized inverse. When computing a generalized inverse we come across two cases:

1.  $\text{rank } A = \text{rank } A_{11}$  where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

And  $A_{11} \in R^{r \times r}$ ,  $A_{12} \in R^{r \times (m-r)}$ ,  $A_{21} \in R^{(n-r) \times r}$ ,  $A_{22} \in R^{(n-r) \times (m-r)}$ . We claim that

$$G = \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

where 0s denote zero matrices of appropriate sizes, is the generalized inverse of A.

2. The upper left  $r \times r$  submatrix of A is **not** invertible.

One way to handle this case is to use permutation matrices P and Q, such that

$$PAQ = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix},$$

$\tilde{A}_{11} \in R^{r \times r}$  and  $\text{rank } \tilde{A}_{11} = r$ . By case 1 generalized inverse of PAQ equals to

$$(PAQ)^g = \begin{bmatrix} \tilde{A}_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

Thus  $(PAQ)(PAQ)^g(PAQ) = PAQ$  holds, Multiplying from the left by  $P^{-1}$  and from the right by  $Q^{-1}$  we get:

$$A(Q(PAQ)^g P)A = A$$

So

$$Q(PAQ)^g P$$

is a **generalized** inverse of A.

**Algorithm** for computing  $A^g$ :

1. Find any nonsingular submatrix B in A of order  $r \times r$ ,
2. in A substitute
  - elements of submatrix B for corresponding elements of  $(B^{-1})^T$ ,
  - all other elements with 0
3. the transpose of the obtained matrix is generalized inverse G

**solutions:**

Let  $A \in R^{n \times m}$  and  $\vec{b} \in R^m$ . If the system  $A\vec{x} = \vec{b}$  is solvable (that is,  $\vec{b} \in C(A)$ ) and G is a generalized inverse of A, then  $\vec{x} = G\vec{b}$  is a solution of the system. Moreover, all solutions of system are exactly vectors of the form

$$x_z = G\vec{b} + (GA - I)z$$

## 1.8 The Moore-Penrose generalized inverse

The MP inverse of  $A \in R^{n \times m}$  is any matrix  $A^+ \in R^{m \times n}$  satisfying the following four conditions:

1.  $A^+$  is a generalized inverse of A:  $AA^+A = A$
2. A is a generalized inverse of  $A^+$ :  $A^+AA^+ = A^+$
3. The square matrix  $AA^+ \in R^{n \times n}$  is symmetric:  $(AA^+)^T = AA^+$

4. The square matrix  $A^+A \in R^{m \times m}$  is symmetric:  $(A^+A)^T = A^+A$

Properties:

- If  $A$  is a square invertible matrix, then it  $A^+ = A^{-1}$
- $((A^+))^+ = A$
- $(A^T)^+ = (A^+)^T$

Construction of the MP inverse (4 cases):

1.  $A^T A \in R^{m \times m}$  is an invertible matrix ( $m \leq n$ )

$$A^+ = (A^T A)^{-1} A^T$$

2.  $AA^T$  is an invertible matrix ( $n \leq m$ )

$$A^+ = A^T (AA^T)^{-1}$$

3.  $\Sigma \in R^{n \times m}$  is diagonal matrix of the form

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}$$

Then the MP inverse is:

$$\Sigma = \begin{bmatrix} \sigma_1^+ & & \\ & \ddots & \\ & & \sigma_n^+ \end{bmatrix}$$

$$\text{where } \sigma_i^+ = \begin{cases} \frac{1}{\sigma_i} & \sigma_i \neq 0 \\ 0 & \sigma_i = 0 \end{cases}$$

4. a general matrix  $A$  (using SVD)

$$A^+ = V \Sigma^+ U^T$$

## 1.9 SVD computation

1. Compute the eigenvalues and an orthonormal basis consisting of eigenvectors of the symmetric matrix  $A^T A$  or  $AA^T$  (depending on which of them is of smaller size)
2. the singular values of the matrix  $A \in R^{n \times m}$  are equal to  $\sigma_i = \sqrt{\lambda_i}$
3. the left singular vectors are the corresponding orthonormal eigenvectors of  $AA^T$
4. the right singular vectors are the corresponding orthonormal eigenvectors of  $A^T A$
5. If  $u$  (resp.  $v$ ) is a left (resp. right) singular vector corresponding to the singular value  $\sigma_i$ , then  $v = Au$  (resp.  $u = A^T v$ ) is a right (resp left) singular vector corresponding to the same singular value

6. the remaining columns of  $U$  (resp.  $V$ ) consist of an orthonormal basis of the kernel (i.e., the eigenspace of  $\lambda = 0$ ) of  $AA^T$  (resp.  $A^T A$ )

## 1.10 General computation of $A^+$

1. For  $A^T A$  compute its **nonzero** eigenvalues  $\lambda_1 \geq \dots \geq \lambda_r > 0$ , and the corresponding orthonormal eigenvectors  $v_1, \dots, v_r$ , and form the matrices:

$$S = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r}) \in R^{r \times r}$$

$$V = [v_1 \dots v_r] \in R^{m \times r}$$

2. Put the vectors

$$u_1 = \frac{Av_1}{\sigma_1}, \dots, u_r = \frac{Av_r}{\sigma_r}$$

in the matrix  $U = [u_1, \dots, u_r]$

3. Then  $A^+ = V \Sigma^+ U^T$

## 1.11 Undetermined systems

Let  $A \in R^{n \times m}$ , where  $m > n$ . A system of equations that has more variables than constraints. Typically such system has infinitely many solutions, but it may happen that it has no solutions  $\rightarrow$  such system is undetermined.

1. An undetermined system of linear equations  $Ax = b$  is solvable iff  $AA^+b = b$ .
2. If there are infinitely many solutions, the solution  $A^+b$  is the one with the smallest norm, i.e

$$\|A^+b\| = \min\{\|x\| : Ax = b\}$$

Moreover, it is the unique solution of smallest norm.

## 1.12 Overdetermined systems

Let  $A \in R^{n \times m}$ , where  $n > m$ . A system of equations that has more constraints than variables. Typically such system has no solutions, but it may happen that it may have one or even infinitely many solutions.

# 2 Nonlinear Models

## 2.1 Definition

Given is a sample of data points  $\{(x_1, y_1), \dots, (x_m, y_m)\} \in R^n \times R$ . We are searching for

$$F : R^n \times R^p \rightarrow R,$$

$$F(x_i, a_i, \dots, a_p) = y_i,$$

$$i = 1, \dots, m$$

which can be nonlinear in parameters  $a_i, \dots, a_p$ . **Examples:** exponential decay/growth, Gaussian model, logistic model,...

We introduce a **vector function**

$$G : R^p \rightarrow R^m, G = (g_1, \dots, g_m)$$

$$g_1(a_1, \dots, a_p) = F(x_1, a_1, \dots, a_p) - y_1$$

$$\vdots$$

$$g_m(a_1, \dots, a_p) = F(x_m, a_1, \dots, a_p) - y_m$$

Solving  $F(x_i, a_i, \dots, a_p) = y_i$ , is equivalent to solving  $G(a_1, \dots, a_p) = (0, \dots, 0)$ . Newton's method will solve the latter system.

## 2.2 Rate of convergence

Let us say that  $e_i$  is the  $i$ -th error of some method and  $\alpha$  is a zero we are searching for. We can show that  $|e_{i+1}| \leq C \cdot \frac{1}{2}|e_i|^r$ , where  $C \in R > 0$  and  $r \in N$ .  $r$  is the rate of convergence of the method.

## 2.3 Case 1 for solving nonlinear equations ( $p = m = 1$ )

$g : R \rightarrow R$ . We have to find a zero of the function  $g$  (let's call it  $\alpha$ ).

### 2.3.1 Tangent method

1. Select an initial approximation  $x_0$ .
2. Iterate  $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0, 1, \dots$  until it converges to  $\alpha$

Problem: the convergence of this method really needs a good initial approximation.

**Theorem:** The sequence  $x_i$  from the tangent method converges with the rate  $r = 2$  if:

- $f'(\alpha) \neq 0$
- $x_0$  was close enough to  $\alpha$

### 2.3.2 Bisection

1. Start with  $x_0, x_1$  such that  $f(x_0)f(x_1) < 0$ .
2. Compute  $x_2 = \frac{x_0 + x_1}{2}$ .
3. Choose  $[x_0, x_2]$  if  $f(x_0)f(x_2) < 0$ , otherwise choose  $[x_2, x_1]$ .
4. Repeat.

Bisection always converges, but it is slow.

Rate of convergence: We can show that  $|e_{i+1}| \approx \frac{1}{2}|e_i|$ . From here we can see that bisection has a linear rate of convergence.

### 2.3.3 Stopping criteria

Iterative method:  $x_0, \dots, x_n, x_n + 1$ . How do we find  $n$  (how many steps of a method do we have to make)? This has to be true:

- $\frac{|x_{n+1} - x_n|}{|x_n|} < \text{tolerance}$
- $|f(x_{n+1})| < \text{tolerance}$

Tolerance is usually  $10^{-10}$ .

## 2.4 case 2 for solving nonlinear equations ( $n = p < 1$ )

### 2.4.1 Newton's or Tangent method

We construct a recursive sequence with:  $x_0$  is an initial term

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

**2.4.x Broyden's method** It approximates  $JF(x^{(m)})$  with some matrix  $B_k$  where:

<div> <div>slope <math>\approx f'(x_1)</math></div> <div><math>\frac{f(x_2) - f(x_1)}{x_2 - x_1} \approx f'(x_1)</math></div> <div><math>f(x_2) - f(x_1) \approx f'(x_1)(x_2 - x_1)</math></div> </div> <div> <p>Where <math>&lt; f(x_2) - f(x_1) &gt;</math> is a vector, <math>&lt; f'(x_1) &gt;</math> is a <math>n \times n</math> matrix <math>B_n</math>, and <math>&lt; (x_2 - x_1) &gt;</math> is a vector.</p> </div>	<div> <div><b>2.4.x Conclusion</b></div> <div> <ol style="list-style-type: none"> <li>1. Use Broyden's method first</li> <li>2. If it doesn't converge, go and compute derivatives for Newton's method.</li> </ol> </div> </div>	<div> <div>(a) Do a few steps of gradient descent(GD)</div> <div>(b) Use the final approximate of GD as entry into Newton's method</div> </div>
--	--	---

3

Curves