Computational Data Mining

Part 1: Linear Algebra

Banded matrices, Least Square

Instructor: Zahra Narimani

Department of Computer Science and Information Technology,
Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

Banded Matrices

- A large proportion of the matrix elements equal to zero
  - e.g., boundary value problems
- Banded Matrice
  - If the nonzero elements are concentrated around the main diagonal

Band Matrix Definition

- A matrix *A* is said to be a *band matrix* if there are natural numbers *p* and *q* such that

$$a_{ij} = 0 \;\text{ if }\; j - i > p \;\text{ or }\; i - j > q$$

- Find p and q?

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} \end{pmatrix}$$

- *bandwidth* of the matrix w = p+q+1

How one can use this property?

- Storing?
- Solving linear systems of equations

Example: *tridiagonal* matrix ($p = q = 1$)

- Stored in three vectors
- Utilize the structure in solving Ax=b
  - How?

LU factorization for a tridiagonal matrix $A_h U = f$

$$\begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & 0 \\ & \ddots & \ddots & \ddots & \\ & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix} =$$

$$\begin{pmatrix} 1 & & & & \\ l_2 & 1 & & 0 & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & 0 & & l_n & 1 \end{pmatrix} \begin{pmatrix} v_1 & c_1 & & & \\ & v_2 & c_2 & & 0 \\ & & \ddots & \ddots & \\ & & & v_{n-1} & c_{n-1} \\ & 0 & & & v_n \end{pmatrix}$$

To determine L, U:

$$\begin{aligned} b_1 = v_1 &\Rightarrow v_1 = b_1 \\ a_k = l_k v_{k-1} &\Rightarrow l_k = a_k / v_{k-1} \\ b_k = l_k c_{k-1} + v_k &\Rightarrow v_k = b_k - l_k c_{k-1}, \quad k = 2, \dots, n \end{aligned}$$

To solve $Ly = f$:

$$\begin{aligned} y_1 &= f_1 \\ l_k y_{k-1} + y_k &= f_k \quad \Rightarrow \quad y_k = f_k - l_k y_{k-1}, \quad k = 2, \ldots, n \end{aligned}$$

To solve $Uu = y$:

$$\begin{aligned} v_n u_n &= y_n \quad \Rightarrow \quad u_n = y_n / v_n \\ v_k u_k + c_k u_{k+1} &= y_k \quad \Rightarrow \quad u_k = (y_k - c_k u_{k+1}) / v_k, \quad k = n-1, \ldots, 1 \end{aligned}$$

operation count: # of multiplications $\sim 3n \ll \frac{n^3}{3}$

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ \gamma_2 & \alpha_2 & \beta_2 & & & \\ & \gamma_3 & \alpha_3 & \beta_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \gamma_{n-1} & \alpha_{n-1} & \beta_{n-1} \\ & & & & \gamma_n & \alpha_n \end{pmatrix}$$

solving Ax=b in tridiagonal matrix

- Assume A is diagonally dominant, so that no pivoting is needed.

```
% LU Decomposition of a Tridiagonal Matrix.
for k=1:n-1
  gamma(k+1)=gamma(k+1)/alpha(k);
  alpha(k+1)=alpha(k+1)*beta(k);
end
% Forward Substitution for the Solution of Ly = b.
y(1)=b(1);
for k=2:n
  y(k)=b(k)-gamma(k)*y(k-1);
end
% Back Substitution for the Solution of Ux = y.
x(n)=y(n)/alpha(n);
for k=n-1:-1:1
  x(k)=(y(k)-beta(k)*x(k+1))/alpha(k);
end
```

- How many operations (multiplications and additions) and divisions?

Operations for computing the LU decomposition

- In the kth step of Gaussian elimination?
- The total number of flops?

Operations for computing the LU decomposition

- In the kth step of Gaussian elimination, one operates on an (n−k+1)×(n−k+1) submatrix,
- The total number of flops:

$$2\sum_{k=1}^{n-1}(n-k+1)^2 \approx \frac{2n^3}{3}$$

Gaussian elimination with partial pivoting

- If A has band width $w = q + p + 1$ (q diagonals under the main diagonal and p over), then, with partial pivoting, the factor U will have band width

$$w_U = p + q + 1$$

- No new nonzero elements will be created in L

Example

$$A = \begin{pmatrix} 4 & 2 & & & \\ 2 & 5 & 2 & & \\ & 2 & 5 & 2 & \\ & & 2 & 5 & 2 \\ & & & 2 & 5 \end{pmatrix} \qquad U = \begin{pmatrix} 2 & 1 & & & \\ & 2 & 1 & & \\ & & 2 & 1 & \\ & & & 2 & 1 \\ & & & & 2 \end{pmatrix}$$

- Cholesky decomposition $U^T U$ and inverse (dense)

$$A^{-1} = \frac{1}{2^{10}} \begin{pmatrix} 341 & -170 & 84 & -40 & 16 \\ -170 & 340 & -168 & 80 & -32 \\ 84 & -168 & 336 & -160 & 64 \\ -40 & 80 & -160 & 320 & -128 \\ 16 & -32 & 64 & -128 & 256 \end{pmatrix}$$

The Least Squares Problem

- Determine the elasticity properties of a spring by attaching different weights to it and measuring its length (Hooke's law)

$$e + \kappa F = l$$

F	1	2	3	4	5
l	7.97	10.2	14.2	16.0	21.2

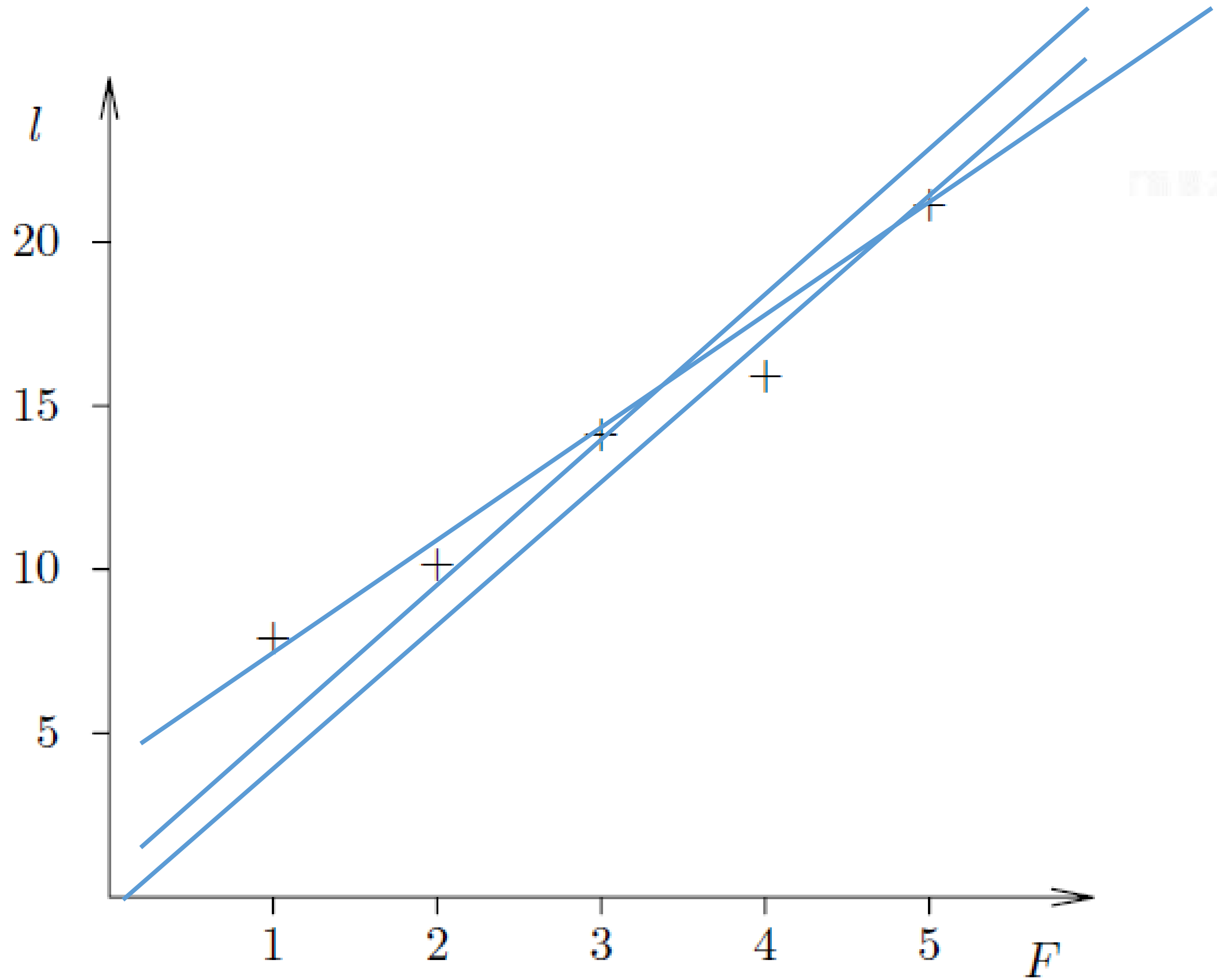
Measurements are subject to error

- Our goal: use all the data in order to minimize the influence of the errors
  - a system with more data than unknowns

$$
\begin{aligned}
e + \kappa 1 &= 7.97,\\
e + \kappa 2 &= 10.2,\\
e + \kappa 3 &= 14.2,\\
e + \kappa 4 &= 16.0,\\
e + \kappa 5 &= 21.2,
\end{aligned}
$$

Matrix form

$$
\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} e \\ \kappa \end{pmatrix} = \begin{pmatrix} 7.97 \\ 10.2 \\ 14.2 \\ 16.0 \\ 21.2 \end{pmatrix}
$$

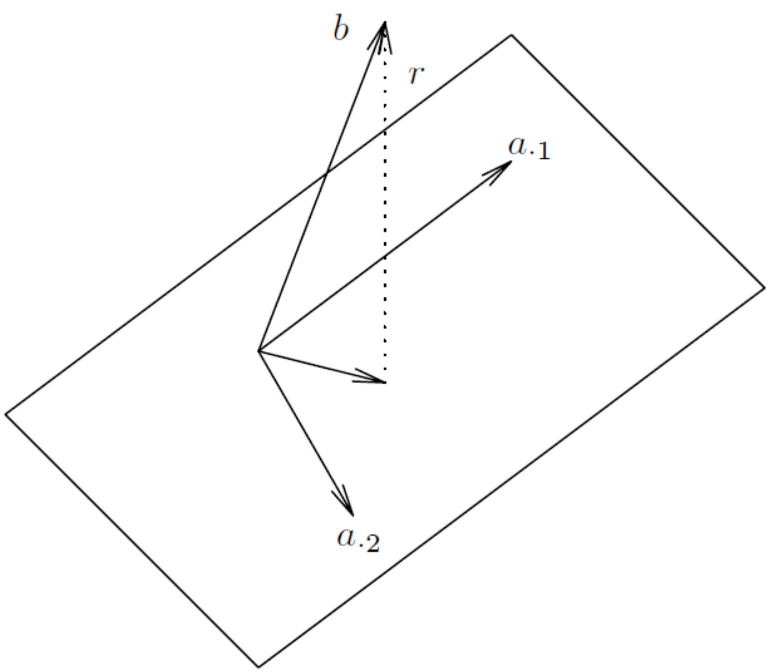

l
20
15
10
5
1
2
3
4
5
F

Overdetermined system

- $A \in \mathbb{R}^{m \times n}, m > n$
- The system *Ax* = *b* is called *overdetermined*
- In general such a system has no solution!

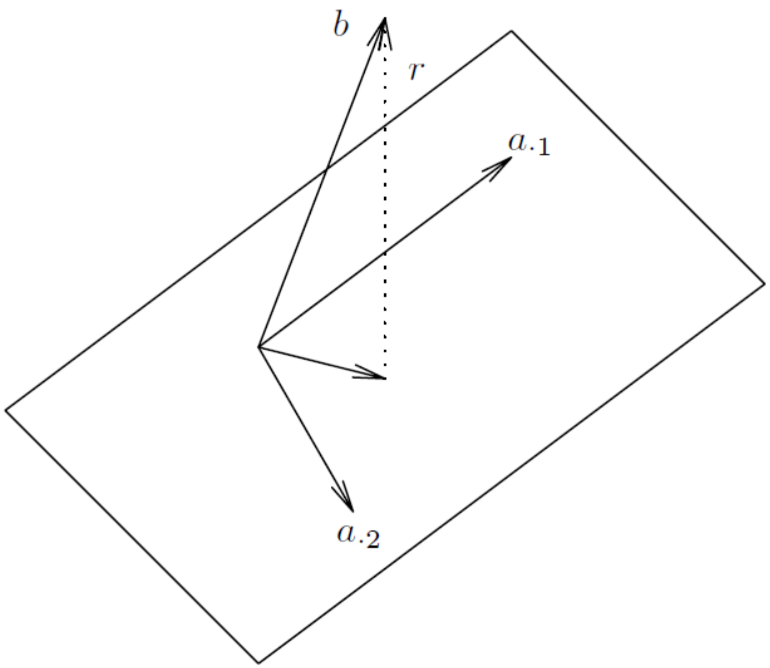
why overdetermined system has no solution? Geometric insight

- Let *m* = 3 and *n* = 2 $x_1.a_1 + x_2.a_2 = b$
- *The residual vector b − Ax is dotted*



Alternative solution

- Let m = 3 and n = 2 $\quad x_1.a_1 + x_1.a_1 = b$
- One alternative to "solving the linear system" is to make the vector $r = b - x_1 a_{\cdot 1} - x_2 a_{\cdot 2} = b - Ax$ as small as possible



Minimizing the residual vector

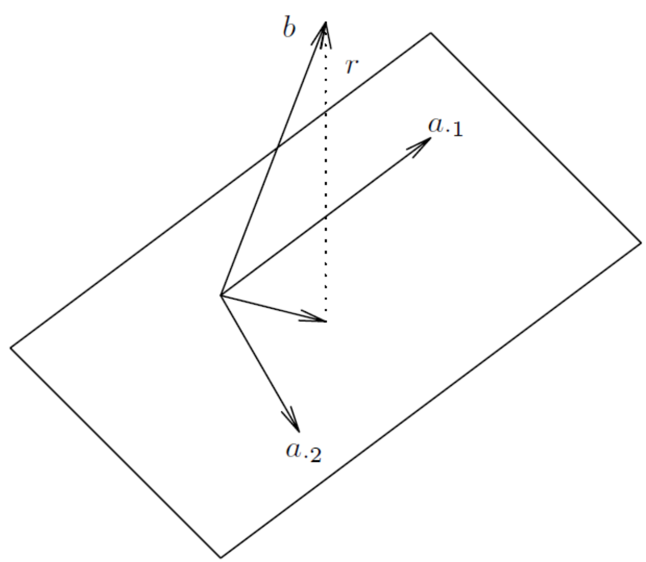
- Solution depends on how we measure the length of the residual vector
- *Least squares method:* standard Euclidean distance
  - Find a vector $x \in \mathbb{R}^n$ that solves the minimization problem:

$$\min_x \| b - Ax \|_2$$

- Since x occurs linearly in this equation, it is called linear least square problem

How to minimize Ax-b?

- Choose the linear combination of vectors in the plane in such a way that the residual vector is orthogonal to the plane



Orthogonality

- We first recall that two nonzero vectors x and y are called orthogonal if $x^T y = 0$ (i.e., cos θ(x, y) = 0).

Making r orthogonal to the plane?

- The columns of the matrix *A* span the plane
- To make r orthogonal to the plane is equivalent to:
  - *making r orthogonal to the columns of A*
- This is valid in general case

$$r^T a_{\cdot j} = 0, \qquad j = 1, 2, \ldots, n$$

$$r^T a_{\cdot j} = 0, \qquad j = 1, 2, \ldots, n$$

$$r^T \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \cdots & a_{\cdot n} \end{pmatrix} = r^T A = 0$$

- Using $r = b - Ax$, we get the normal equations for determining the coefficients in x.

$$A^T A x = A^T b$$

Theorem 3.10

If the column vectors of A are linearly independent, then the normal equations

$$A^T A x = A^T b$$

are nonsingular and have a unique solution.

Proof:

- $A^T A$ *is non-singular* ($y^T y = \sum_i y_i^2$)
- Normal equations have a unique solution $\hat{x}$ *and* for all $r = b - Ax$

$$||\hat{r}||_2 \leq ||r||_2$$

$\|\hat{r}\|_2 \leq \|r\|_2$ for all r = b−Ax

$$r = b - A\widehat{x} + A(\widehat{x} - x) = \widehat{r} + A(\widehat{x} - x)$$

$$\begin{aligned}\|r\|_2^2 = r^T r &= (\widehat{r} + A(\widehat{x} - x))^T(\widehat{r} + A(\widehat{x} - x)) \\ &= \widehat{r}^T\widehat{r} + \widehat{r}^T A(\widehat{x} - x) + (\widehat{x} - x)^T A^T \widehat{r} + (\widehat{x} - x)^T A^T A(\widehat{x} - x).\end{aligned}$$

Since $A^T\widehat{r} = 0$, the two terms in the middle are equal to zero

$$\|r\|_2^2 = \widehat{r}^T\widehat{r} + (\widehat{x} - x)^T A^T A(\widehat{x} - x) = \|\widehat{r}\|_2^2 + \|A(\widehat{x} - x)\|_2^2 \geq \|\widehat{r}\|_2^2,$$

which was to be proved.

Solve the equation for spring elasticity

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} e \\ \kappa \end{pmatrix} = \begin{pmatrix} 7.97 \\ 10.2 \\ 14.2 \\ 16.0 \\ 21.2 \end{pmatrix} \Rightarrow A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix}, \quad b = \begin{pmatrix} 7.97 \\ 10.2 \\ 14.2 \\ 16.0 \\ 21.2 \end{pmatrix}$$

Solve the equation for spring elasticity

$$A^T A x = A^T b$$

Using MATLAB we then get

```
>> C=A'*A              % Normal equations

C =   5     15
     15     55

>> x=C\(A'*b)

x = 4.2360
    3.2260
```

x = A\B solves the system of linear equations A*x = B.

Drawbacks

- Forming $A^T A$ can lead to loss of information
- The condition number $A^T A$ is the square of that of A:

$$\kappa(A^T A) = (\kappa(A))^2$$

Drawback - Loss of information

$$A = \begin{pmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{pmatrix} \Rightarrow A^T A = \begin{pmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{pmatrix}$$

- If is ϵ *is* so small that the floating point representation of $1 + \epsilon^2$ satisfies $\mathrm{fl}[1+\epsilon^2] = 1$, then in floating point arithmetic the normal equations become singular.
- Thus vital information that is present in A is lost in forming $A^T A$.

Drawbacks - condition number of $A^T A$

- **Condition number:** a function with respect to an argument measures how much the output value of the function can change for a small change in the input argument.
  - Used to measure how sensitive a function is to changes or errors in the input, and how much error in the output results from an error in the input.
- A problem with a low condition number is said to be **well-conditioned**, while a problem with a high condition number is said to be **ill-conditioned**.

Ill-Conditioned (example)

- A mathematical problem or series of equations is ill-conditioned if a small change in the independent variable (input) leads to a large change in the dependent variable (output)

$$\begin{cases} x_1 + 2x_2 = 10 \\ 1.1x_1 + 2x_2 = 10.4 \end{cases}$$

$$x_1 = \frac{2(10) - 2(10.4)}{1(2) - 2(1.1)} = 4 \qquad x_2 = \frac{1(10.4) - 1.1(10)}{1(2) - 2(1.1)} = 3$$

$$\boxed{a_{21} \text{ from } 1.1 \text{ to } 1.05}$$

$$x_1 = \frac{2(10) - 2(10.4)}{1(2) - 2(1.05)} = 8 \qquad x_2 = \frac{1(10.4) - 1.1(10)}{1(2) - 2(1.05)} = 1$$

Ill-Conditioned

$$\boxed{\kappa(A^T A) = (\kappa(A))^2}$$

```
A =   1     1
      1     2
      1     3
      1     4
      1     5

cond(A) = 8.3657

cond(A'*A) = 69.9857
```

```
A =   1   101
      1   102
      1   103
      1   104
      1   105

cond(A) = 7.5038e+03

cond(A'*A) = 5.6307e+07
```

Any Question?