UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical &
Computer Engineering

ECE 204 *Numerical methods*

Least-squares best-fitting polynomials

Douglas Wilhelm Harder, LEL, M.Math.
dwharder@uwaterloo.ca
dwharder@gmail.com

Introduction

- In this topic, we will
  - Describe approximating $A\mathbf{u} = \mathbf{v}$ when no exact solution exists
  - Introduce the normal equation
  - Use the normal equation to find the
    - Best-fitting linear polynomial that fits noisy data
    - Best-fitting quadratic polynomial that fits noisy data

Review

- From linear algebra:
  - Suppose that $A:\mathbf{R}^2 \rightarrow \mathbf{R}^4$ and $A\mathbf{u} = \mathbf{v}$ has no solution
  - If the columns of A are linearly independent, this requires that the system is overdetermined with rank 2
    - That is, there are more equations than unknowns and the system is inconsistent

$$\begin{pmatrix} 3.1 & 4.0 \\ 2.8 & 7.6 \\ 5.9 & 1.2 \\ 6.4 & 8.7 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 5.6 \\ 0.8 \\ 7.3 \\ 9.1 \end{pmatrix}$$

Review

- From linear algebra:
  - A simple example:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 8 \end{pmatrix}$$

  - The first two rows indicate that $u_1 = u_2 = 1$
  - In this case, however, $2u_1 + 3u_2 = 5$, and not 8
  - Thus, this system of three linear equations in two unknowns is inconsistent
    - No solution exists

Review

- Thus, it is almost certain that no linear combination of the columns of *A* that equals the target vector
  - Thus, for all u_1 and u_2,

$$u_1 \begin{pmatrix} 3.1 \\ 2.8 \\ 5.9 \\ 6.4 \end{pmatrix} + u_2 \begin{pmatrix} 4.0 \\ 7.6 \\ 1.2 \\ 8.7 \end{pmatrix} \neq \begin{pmatrix} 5.6 \\ 0.8 \\ 7.3 \\ 9.1 \end{pmatrix}$$

- What's the next-best choice?
  - How about, what linear combination is *closest to* **v?**

Review

- Let $A: \mathscr{U} \to \mathscr{V}$ be a linear map
  - Usually, $A: \mathbf{R}^n \to \mathbf{R}^m$
- Thus, we want to minimize $\|A\mathbf{u} - \mathbf{v}\|_2$
  - That is, find a **u** that makes this as small as possible
  - Now, consider a plane (e.g., a floor) and a point not on that plane
    - The location on the plane closest to the point is one that forms a perpendicular line
  - We need to find a vector $\mathbf{u}_0$ such that $A\mathbf{u}_0 - \mathbf{v}$ is perpendicular to all vectors in the range of $A\mathbf{u}$
    - Two vectors are perpendicular if their dot product is zero

Review

- Thus, we require that

$$(A\mathbf{u}_0 - \mathbf{v}) \cdot A\mathbf{u} = 0$$

for all $\mathbf{u}$ in the domain $\mathbf{R}^n$

$$\mathbf{v} \cdot (A\mathbf{u}) = (A^{\mathrm{T}}\mathbf{v}) \cdot \mathbf{u}$$

  - If this is true, then

$$A^{\mathrm{T}}(A\mathbf{u}_0 - \mathbf{v}) \cdot \mathbf{u} = 0$$

  must be true for all $\mathbf{u}$ in $\mathbf{R}^n$

  - This is true if and only if

$$A^{\mathrm{T}}(A\mathbf{u}_0 - \mathbf{v}) = \mathbf{0}$$

$$A^{\mathrm{T}}A\mathbf{u}_0 - A^{\mathrm{T}}\mathbf{v} = \mathbf{0}$$

$$A^{\mathrm{T}}A\mathbf{u}_0 = A^{\mathrm{T}}\mathbf{v}$$

$$\mathbf{u}_0 = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\mathbf{v}$$

Review

- Therefore, the solution $\mathbf{u}_0$ to

$$A^{\mathrm{T}} A \mathbf{u}_0 = A^{\mathrm{T}} \mathbf{v}$$

which may sometimes be calculated as

$$\mathbf{u}_0 = \left(A^{\mathrm{T}} A \right)^{-1} A^{\mathrm{T}} \mathbf{v}$$

is that vector $\mathbf{u}_0$ that minimizes

$$\left\| A \mathbf{u}_0 - \mathbf{v} \right\|_2$$

Review

- Let's try this out:

```
>> A = [3.1 4.0; 2.8 7.6; 5.9 1.2; 6.4 8.7];
>> v = [5.6 0.8 7.3 9.1]';
>> u = (A'*A) \ A'*v
    u =
        1.472633619403234
       -0.169731501459659
>> A*u
    ans =
        3.886238214311391
        2.833414723235649
        8.484860552727493
        7.948191101481669
>> norm( A*u - v )
    ans =
        3.130864603088711
```

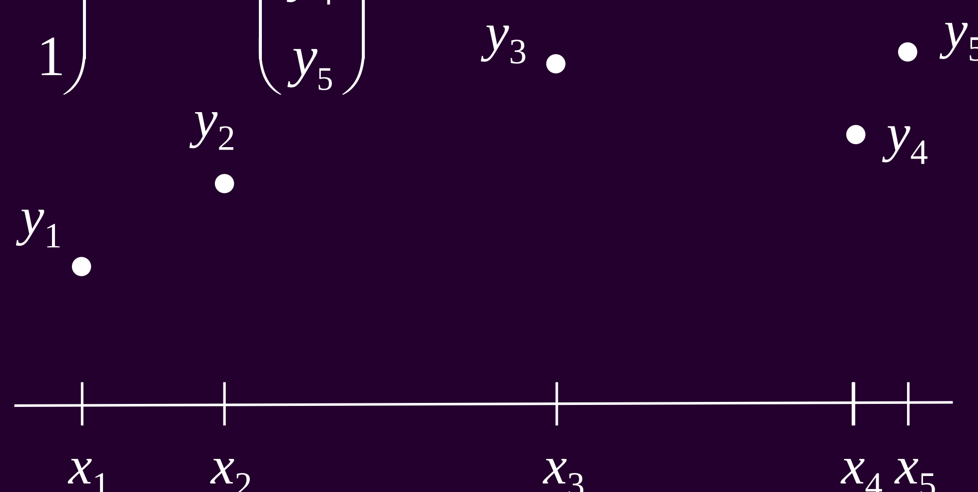
$$\begin{pmatrix} 3.1 & 4.0 \\ 2.8 & 7.6 \\ 5.9 & 1.2 \\ 6.4 & 8.7 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 5.6 \\ 0.8 \\ 7.3 \\ 9.1 \end{pmatrix}$$

$$\begin{pmatrix} 3.886238214311391 \\ 2.833414723235649 \\ 8.484860552727493 \\ 7.948191101481669 \end{pmatrix}$$

Least-squares best-fitting linear polynomial

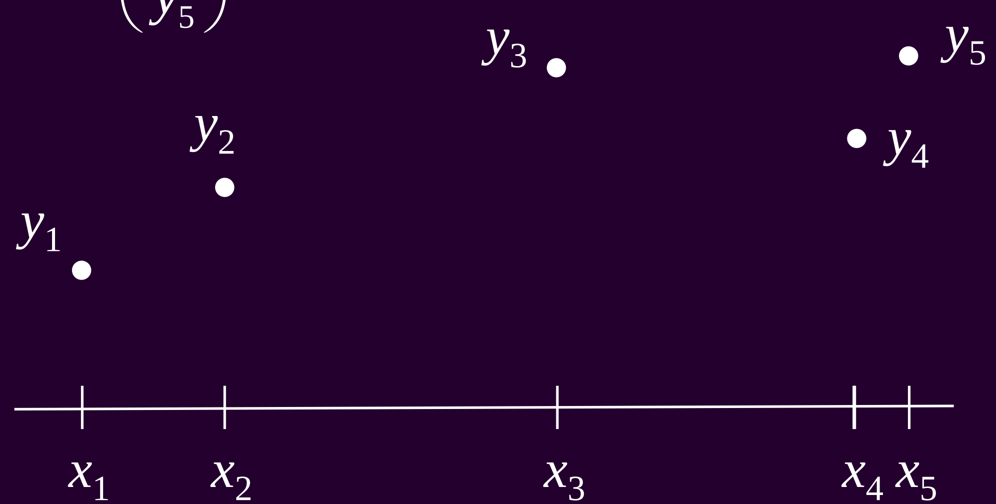
- Suppose we have some data points
  - If there were only two points, we could solve $\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$
  - Now, however, we have five:

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ x_5 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$



Least-squares best-fitting linear polynomial

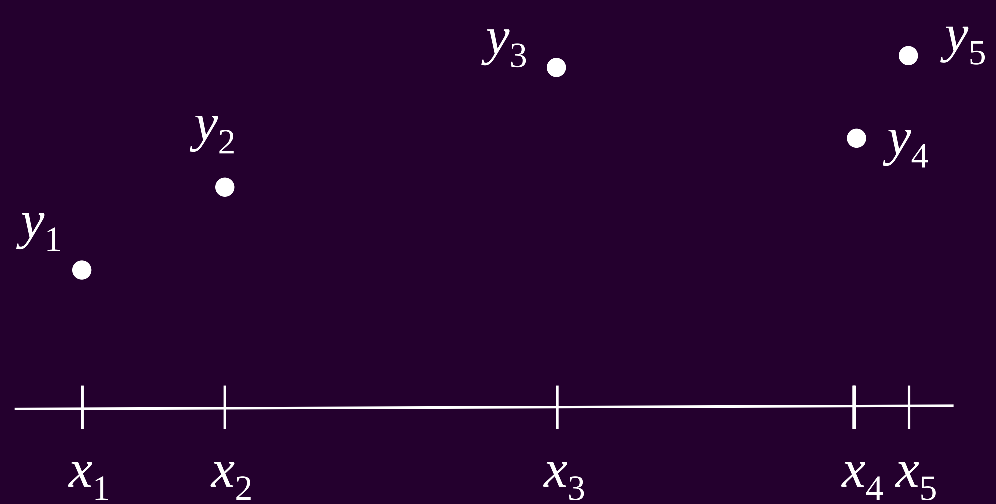
- We proceed by asking:
  - What linear combination $a_1 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} + a_0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ comes closest to our target vector $\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$



Least-squares best-fitting linear polynomial

- We have already seen the solution:
  - Solve $A^{\mathrm{T}}A\mathbf{a} = A^{\mathrm{T}}\mathbf{y}$

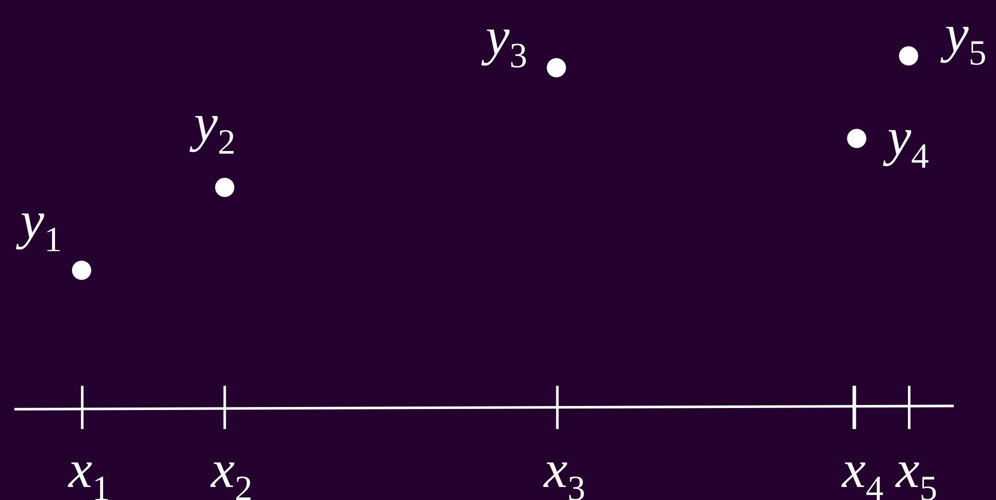
$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ x_5 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$



Least-squares best-fitting linear polynomial

- We have already seen the solution:
  - Solve $A^{\mathrm{T}} A\mathbf{a} = A^{\mathrm{T}}\mathbf{y}$

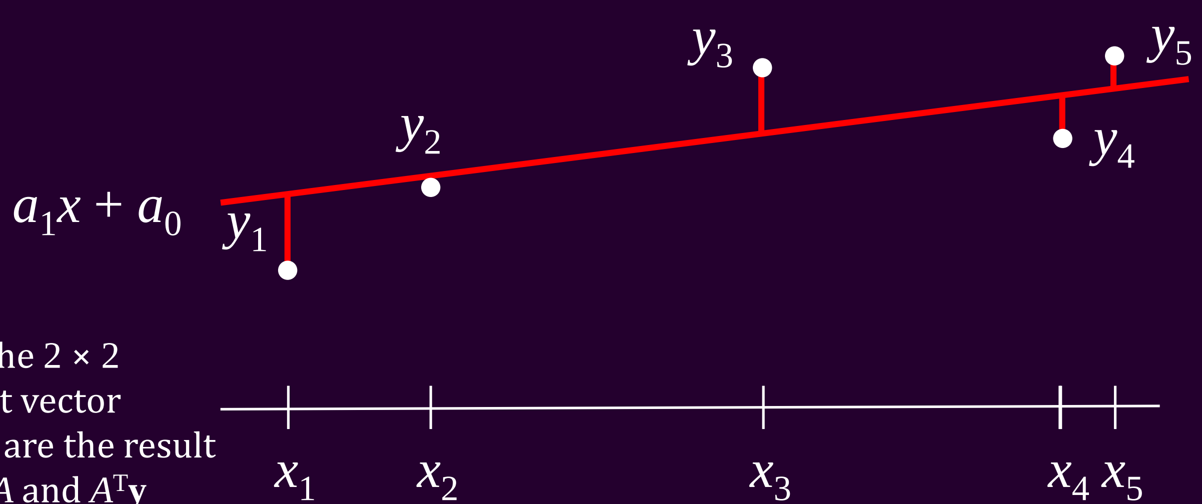
$$\begin{pmatrix} \sum_{k=1}^{n} x_k^2 & \sum_{k=1}^{n} x_k \\ \sum_{k=1}^{n} x_k & n \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} x_k y_k \\ \sum_{k=1}^{n} y_k \end{pmatrix}$$



Least-squares best-fitting linear polynomial

- The solution gives us the best-fitting line
  - The sum of the squares of the errors is minimized

$$\begin{pmatrix} \sum_{k=1}^{n} x_k^2 & \sum_{k=1}^{n} x_k \\ \sum_{k=1}^{n} x_k & n \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} x_k y_k \\ \sum_{k=1}^{n} y_k \end{pmatrix}$$

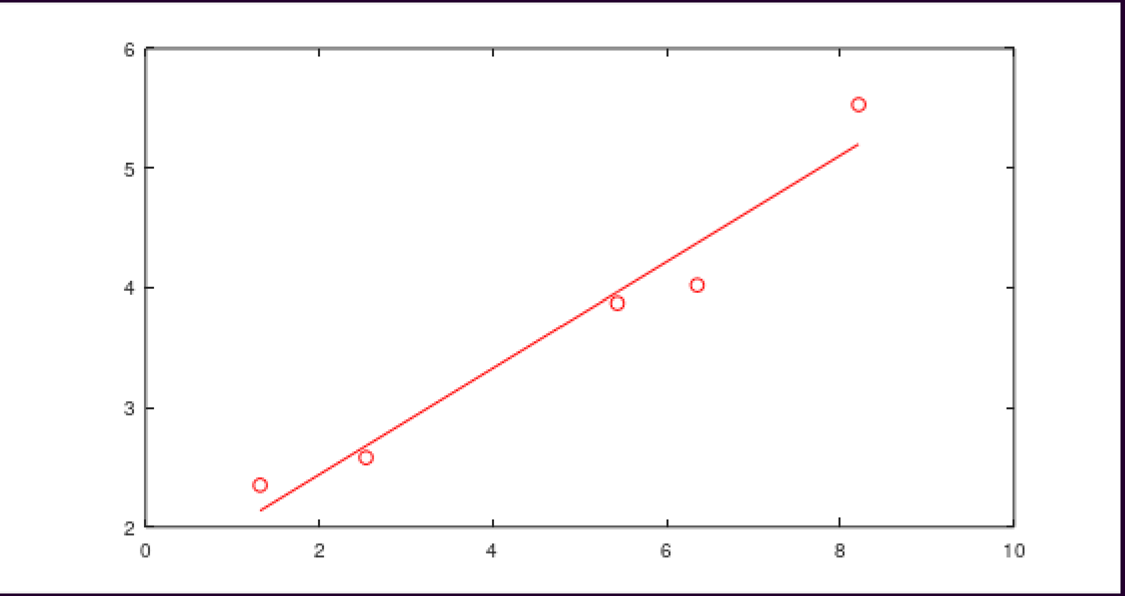


Do not memorize the 2 × 2 matrix or the target vector
- Understand they are the result of calculating A^TA and $A^T\mathbf{y}$

Example

- Let's try this in MATLAB

```
>> x = [1.32 2.54 5.43 6.35 8.21]';
>> y = [2.35 2.58 3.87 4.02 5.53]';
>> plot( x, y, 'ro' )
>> A = vander( x, 2 )
 A =
      1.3200   1.0000
      2.5400   1.0000
      5.4300   1.0000
      6.3500   1.0000
      8.2100   1.0000
>> format long
>> a = (A'*A) \ (A'*y)
 a =
      0.444616162573875
      1.549180904522615
>> hold on
>> plot( x, polyval( a, x ), 'r' );
```



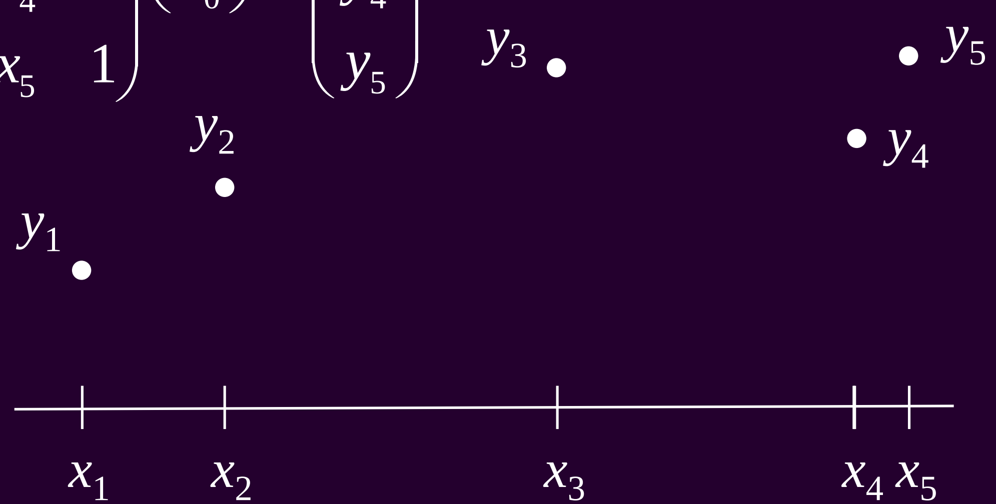
This Matlab code is provided for demonstration purposes and is not required for the examination.

Least-squares best-fitting quadratic polynomial

- Suppose we have some data points
  - If there were three points, we could solve
  - Now, however, we have five:

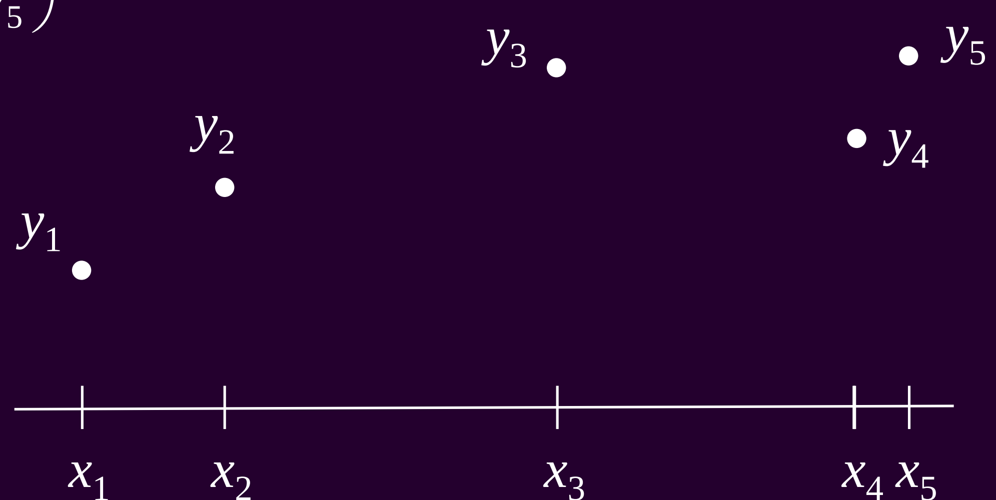
$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix}\begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ x_4^2 & x_4 & 1 \\ x_5^2 & x_5 & 1 \end{pmatrix}\begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$



Least-squares best-fitting quadratic polynomial

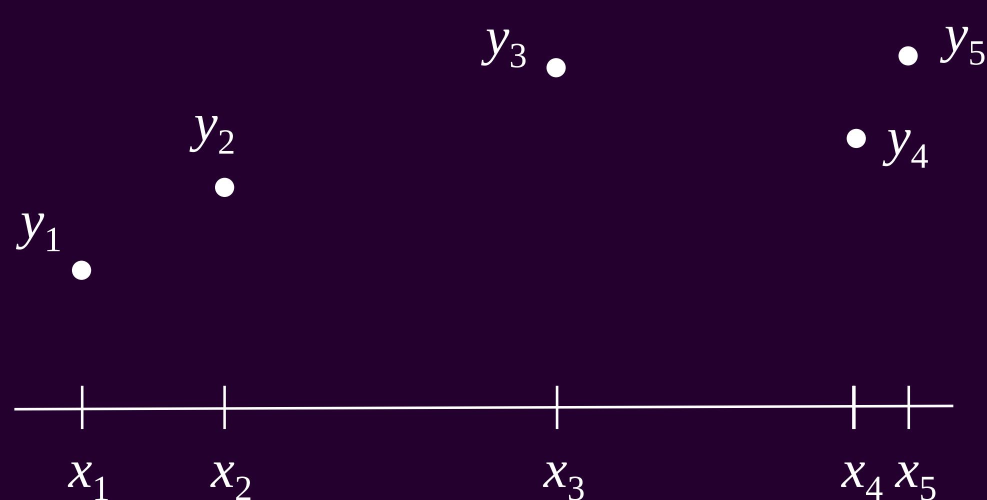
- We proceed by asking:
  - What linear combination $a_2\begin{pmatrix}x_1^2\\x_2^2\\x_3^2\\x_4^2\\x_5^2\end{pmatrix}+a_1\begin{pmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{pmatrix}+a_0\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}$ comes closest to $\mathbf{y}=\begin{pmatrix}y_1\\y_2\\y_3\\y_4\\y_5\end{pmatrix}$



Least-squares best-fitting quadratic polynomial

- As before, we can solve this:
  - Solve $A^{\mathrm{T}} A\mathbf{a} = A^{\mathrm{T}}\mathbf{y}$

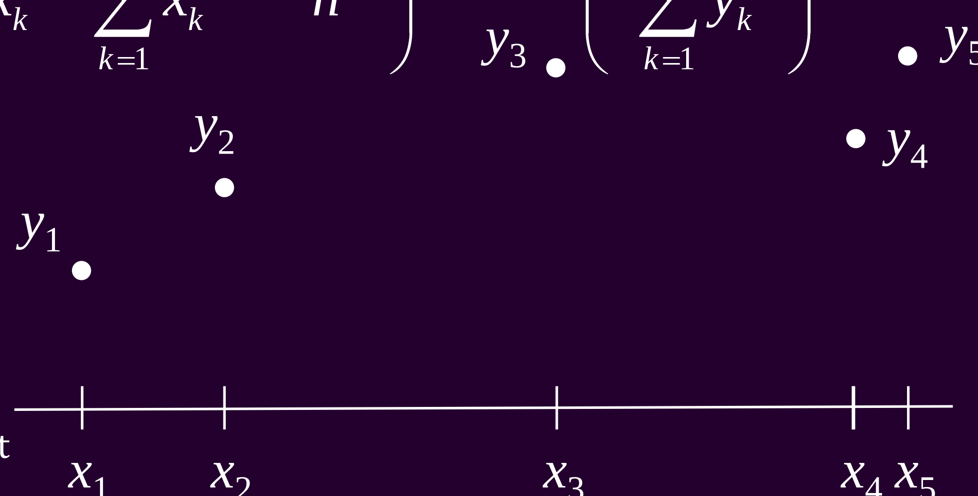
$$\begin{pmatrix} x_1^2 & x_2^2 & x_3^2 & x_4^2 & x_5^2 \\ x_1 & x_2 & x_3 & x_4 & x_5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ x_4^2 & x_4 & 1 \\ x_5^2 & x_5 & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} x_1^2 & x_2^2 & x_3^2 & x_4^2 & x_5^2 \\ x_1 & x_2 & x_3 & x_4 & x_5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$



Least-squares best-fitting quadratic polynomial

- As before, we can solve this:
  - Solve $A^{\mathrm{T}} A\mathbf{a} = A^{\mathrm{T}}\mathbf{y}$

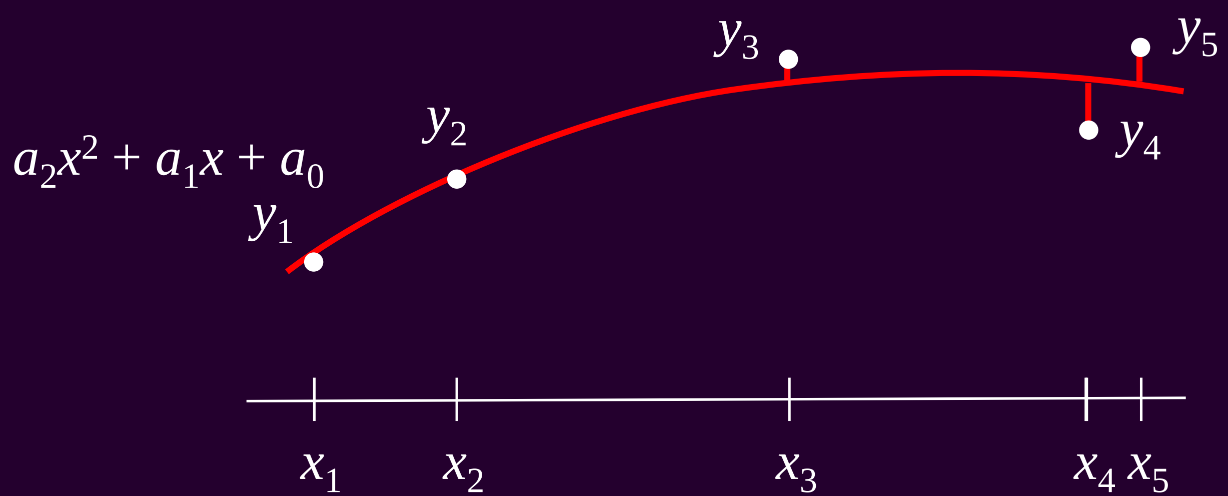
$$\begin{pmatrix} \sum_{k=1}^{n} x_k^4 & \sum_{k=1}^{n} x_k^3 & \sum_{k=1}^{n} x_k^2 \\ \sum_{k=1}^{n} x_k^3 & \sum_{k=1}^{n} x_k^2 & \sum_{k=1}^{n} x_k \\ \sum_{k=1}^{n} x_k^2 & \sum_{k=1}^{n} x_k & n \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} x_k^2 y_k \\ \sum_{k=1}^{n} x_k y_k \\ \sum_{k=1}^{n} y_k \end{pmatrix}$$



Do not memorize the 3 × 3 matrix or the target vector
- Understand they are the result of calculating $A^{\mathrm{T}}A$ and $A^{\mathrm{T}}\mathbf{y}$

Least-squares best-fitting quadratic polynomial

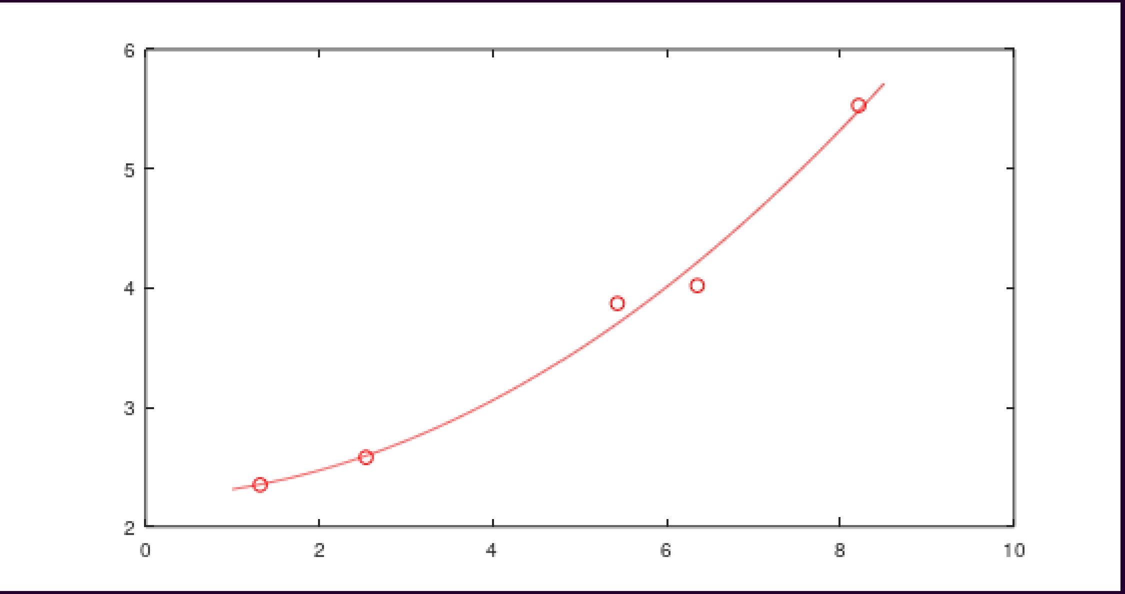
- The solution gives us a quadratic curve that most closely approximates these points
  - The sum of the squares of the errors is minimized



Example

- Let's try this in MATLAB

```
>> x = [1.32 2.54 5.43 6.35 8.21]';
>> y = [2.35 2.58 3.87 4.02 5.53]';
>> plot( x, y, 'ro' )
>> A = vander( x, 3 )
    A =
       1.7424   1.32   1
       6.4516   2.54   1
      29.4849   5.43   1
      40.3225   6.35   1
      67.4041   8.21   1
>> format long
>> a = (A'*A) \ (A'*y)
    a =
         0.04547642859987164
         0.02113915928445630
         2.24661642457417
>> hold on
>> xs = 1:0.01:8.5;
>> plot( xs, polyval( a, xs ), 'r' );
```



This Matlab code is provided for demonstration purposes and is not required for the examination.

Example

- Incidentally, in MATLAB if you try to solve an overdetermined system of linear equations, it automatically gives you the least-squares best-fitting solution:

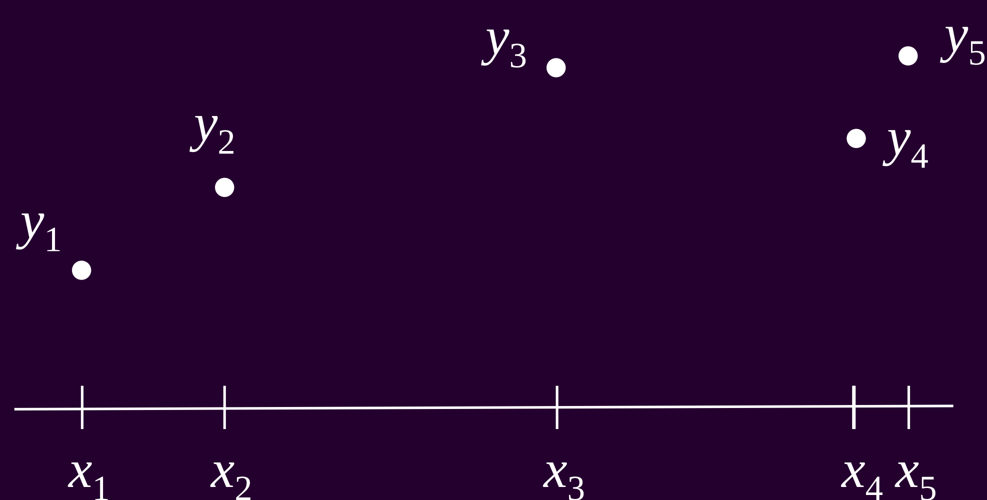
```
>> a = (A'*A) \ (A'*y)
    a =
        0.04547642859987164
        0.02113915928445630
        2.246661642457417
>> a = A \ y
    a =
        0.04547642859987018
        0.02113915928447047
        2.246661642457392
```

$$\mathbf{a} = \begin{pmatrix} 0.04547642859987033 \\ 0.02113915928446891 \\ 2.246661642457394 \end{pmatrix}$$

Least-squares best-fitting constant polynomial

- What is the best constant polynomial $y = a_0$ passing through data?

$$\begin{pmatrix}1 & 1 & 1 & 1 & 1\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}\begin{pmatrix}a_0\end{pmatrix} = \begin{pmatrix}1 & 1 & 1 & 1 & 1\end{pmatrix}\begin{pmatrix}y_1\\y_2\\y_3\\y_4\\y_5\end{pmatrix}$$

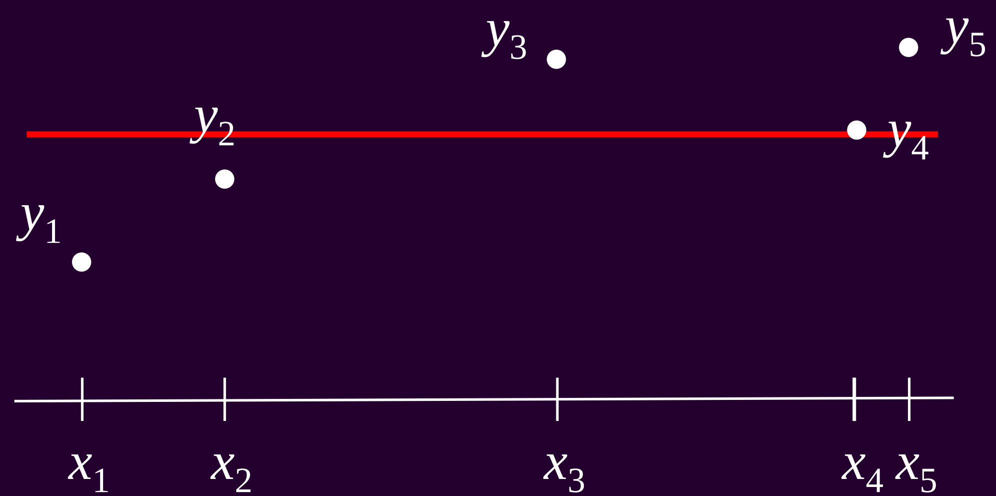


Least-squares best-fitting constant polynomial

- What is the best constant polynomial $y = a_0$ passing through data?

$$(n)(a_0) = \left(\sum_{k=1}^{n} y_k \right)$$

  – The solution is $a_0 = \frac{1}{n} \sum_{k=1}^{n} y_k$



Summary

- Following this topic, you now
  - Understand the idea of finding the solution such that $A\mathbf{u}$ is closest to a target vector $\mathbf{v}$
  - Know that this requires you to solve $A^{\mathrm{T}}A\mathbf{u} = A^{\mathrm{T}}\mathbf{v}$
  - Understand that this can be used to find least-squares best-fitting polynomials passing through data
    - We can find a least-squares constant polynomial, least-squares linear polynomial, least-squares quadratic polynomial and others

References

[1] https://en.wikipedia.org/wiki/Least_squares

Acknowledgments

None so far.

Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`. Mathematical equations are prepared in MathType by Design Science, Inc.

Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

Disclaimer

These slides are provided for the ECE 204 *Numerical methods* course taught at the University of Waterloo. The material in it reflects the author's best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.