

Lecture 6

Linear Regression

Virginia Tech Department of Computer Science
Fall 2026 • Sumeet Khatri

Linear systems and exact solutions

A linear system asks for a vector x satisfying $Ax = b$.

Gaussian elimination

Row operations reduce the augmented matrix to a triangular system.

Back substitution recovers the unknowns.
Pivots reveal consistency and free variables.

The inverse

For square, invertible A , each response vector gives a unique solution.

$$x = A^{-1}b$$

A singular or rectangular matrix has no two-sided inverse.

The pseudoinverse and least squares

For $\text{rank}(A) = r > 0$, the compact SVD gives

$$A = U_r \Sigma_r V_r^T, \quad A^+ = V_r \Sigma_r^{-1} U_r^T$$

The pseudoinverse selects a least-squares solution of minimum coefficient norm.

$$\mathbf{x}_* = A^+ \mathbf{b}, \quad \hat{\mathbf{b}} = AA^+ \mathbf{b}$$

Today we prove this result in detail and use it to fit lines and curves.

Fitting a line to data

We have data of the form $(t_1, b_1), (t_2, b_2), \dots, (t_m, b_m)$. To fit this data to a line, we formulate a matrix equation as follows.

Rows are observations. Columns are adjustable coefficients. Inputs are known; intercept and slope are unknown.

$$f(t) = \alpha + \beta t, \quad b_i \approx \alpha + \beta t_i$$

$$A = \begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

The matrix A is the *design matrix*.

“Linear” describes the coefficients

Choose the *feature functions* first. They may be non-linear in the input.

$$f(t) = \sum_{j=0}^d x_j \phi_j(t), \quad A_{i,j+1} = \phi_j(t_i)$$

Model	Fitted function	Design row
Line	$x_0 + x_1 t$	$(1, t_i)$
Quadratic	$x_0 + x_1 t + x_2 t^2$	$(1, t_i, t_i^2)$
Wave	$x_0 + x_1 \sin t + x_2 \cos t$	$(1, \sin t_i, \cos t_i)$

The coefficients x_j are to be optimized

What exactly are we minimizing?

Residual = observed response minus predicted response. Least squares chooses coefficients to minimize the sum of squared residuals/errors (SSE).

$$\hat{\mathbf{b}} = A\mathbf{x}, \quad \mathbf{r} = \mathbf{b} - A\mathbf{x}$$

Linear regression problem: minimize $\text{SSE}(\mathbf{x}) = \sum_{i=1}^m r_i^2 = \|\mathbf{b} - A\mathbf{x}\|_2^2$ over vectors $\mathbf{x}$

$$\text{MSE} = \frac{\text{SSE}}{m}, \quad \text{RMSE} = \sqrt{\frac{\text{SSE}}{m}}$$

(Mean squared error) (Root mean squared error)

Theorem: the pseudoinverse solves linear/least-squares regression

For $A \in \mathbb{R}^{m \times p}$, least-squares solutions always exist. The pseudoinverse selects the unique one with smallest coefficient norm.

$$\mathbf{x}_* = A^+ \mathbf{b}, \quad \mathbf{x} = A^+ \mathbf{b} + \mathbf{z}, \quad \mathbf{z} \in \text{Null}(A)$$

$$\mathbf{r}_* = (I_m - H) \mathbf{b}, \quad H = AA^+$$

When $\text{Null}(A) = \{\mathbf{0}\}$, the pseudoinverse gives a unique coefficient vector.

Proof: recall the SVD

$$A = U_r \Sigma_r V_r^\top, \quad A^+ = V_r \Sigma_r^{-1} U_r^\top$$

$$\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r), \quad \sigma_j > 0$$

$$U_r^\top U_r = I_r, \quad V_r^\top V_r = I_r$$

$$A : m \times p \quad U_r : m \times r \quad \Sigma_r, \Sigma_r^{-1} : r \times r \quad V_r : p \times r$$

Proof: the unknown coefficients and objective

Fix $A \in \mathbb{R}^{m \times p}$ and $\mathbf{b} \in \mathbb{R}^m$. The unknown is $\mathbf{x} \in \mathbb{R}^p$.

$$\min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}), \quad F(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2^2 = \sum_{i=1}^m \left(b_i - \sum_{k=1}^p A_{ik}x_k \right)^2$$

$$\frac{\partial F}{\partial x_k} = 2 \sum_{i=1}^m A_{ik} \left(\sum_{\ell=1}^p A_{i\ell}x_{\ell} - b_i \right), \quad k = 1, \dots, p$$

$$\nabla F(\mathbf{x}) = 2A^{\top}(A\mathbf{x} - \mathbf{b}) = \mathbf{0} \quad \implies \quad A^{\top}A\mathbf{x} = A^{\top}\mathbf{b}$$

Zero gradient is necessary for a minimum. We will prove global optimality after solving these equations.

Proof: solving the normal equations

For $r > 0$, the compact SVD turns the normal equations into equations for the row-space coordinates.

$$\begin{aligned}
 V_r \Sigma_r^2 V_r^\top \mathbf{x} &= V_r \Sigma_r U_r^\top \mathbf{b} \\
 \implies \Sigma_r^2 V_r^\top \mathbf{x} &= \Sigma_r U_r^\top \mathbf{b} && \text{(multiply by } V_r^\top) \\
 \implies V_r^\top \mathbf{x} &= \Sigma_r^{-1} U_r^\top \mathbf{b} && \text{(multiply by } \Sigma_r^{-2})
 \end{aligned}$$

A vector with exactly those coordinates is:

$$\begin{aligned}
 \mathbf{x}_* &= V_r \Sigma_r^{-1} U_r^\top \mathbf{b} = A^+ \mathbf{b} \\
 V_r^\top \mathbf{x}_* &= (V_r^\top V_r) \Sigma_r^{-1} U_r^\top \mathbf{b} = \Sigma_r^{-1} U_r^\top \mathbf{b}
 \end{aligned}$$

Proof: the candidate gives a global minimum

Compare the candidate with any other coefficient vector.

$$\mathbf{r}_* = \mathbf{b} - A\mathbf{x}_*, \quad A^\top \mathbf{r}_* = \mathbf{0}, \quad \mathbf{h} \in \mathbb{R}^p$$

$$\begin{aligned} F(\mathbf{x}_* + \mathbf{h}) &= \|\mathbf{r}_* - A\mathbf{h}\|_2^2 \\ &= \|\mathbf{r}_*\|_2^2 - 2\mathbf{h}^\top A^\top \mathbf{r}_* + \|A\mathbf{h}\|_2^2 \\ &= F(\mathbf{x}_*) + \|A\mathbf{h}\|_2^2 \\ &\geq F(\mathbf{x}_*) \end{aligned}$$

The cross term vanishes. Every competitor has at least the candidate's residual error. This proves global optimality!

Proof: all least-squares minimizers

The expansion tells us exactly when a different coefficient vector is equally good.

$$F(\mathbf{x}_* + \mathbf{h}) = F(\mathbf{x}_*) \iff A\mathbf{h} = \mathbf{0}$$

$$\mathbf{x} = A^+\mathbf{b} + \mathbf{z}, \quad \mathbf{z} \in \text{Null}(A)$$

$$A(\mathbf{x}_* + \mathbf{z}) = A\mathbf{x}_* + A\mathbf{z} = A\mathbf{x}_*$$

Every minimizer gives the same fitted data, residual, and minimum error. Coefficients are unique precisely when the null space contains only the zero vector.

Proof: minimum norm among the minimizers

Least squares is already solved. Now minimize coefficient size only among its minimizers.

$$Az = \mathbf{0} \implies \Sigma_r V_r^\top z = \mathbf{0} \implies V_r^\top z = \mathbf{0}$$

$$\mathbf{x}_*^\top z = (\Sigma_r^{-1} U_r^\top \mathbf{b})^\top V_r^\top z = 0$$

$$\|\mathbf{x}_* + z\|_2^2 = \|\mathbf{x}_*\|_2^2 + \|z\|_2^2 \geq \|\mathbf{x}_*\|_2^2$$

Equality requires a zero null-space component. The pseudoinverse gives the unique minimum-norm choice among the least-squares minimizers.

Consequence: the normal equations

A coefficient vector minimizes the squared residual exactly when it satisfies the normal equations.

$$A^T(\mathbf{b} - A\mathbf{x}_*) = \mathbf{0} \quad \Longleftrightarrow \quad A^T A \mathbf{x}_* = A^T \mathbf{b}$$

$$\text{rank}(A) = p \quad \Longrightarrow \quad \mathbf{x}_* = (A^T A)^{-1} A^T \mathbf{b}$$

Fitted data are always unique. Coefficients are unique exactly when the columns are independent. (Full column rank; equivalent to $\text{Null}(A) = \{\mathbf{0}\}$).

Four different uniqueness questions

Training fitted values: always unique.

Coefficients: unique exactly when the columns are independent.

New-input prediction: depends on the input when columns are dependent.

Minimum-norm choice: always unique in fixed coordinates.

$$\hat{\mathbf{b}} = AA^+\mathbf{b}, \quad \mathbf{x} = A^+\mathbf{b} + \mathbf{z}, \quad \mathbf{z} \in \text{Null}(A)$$

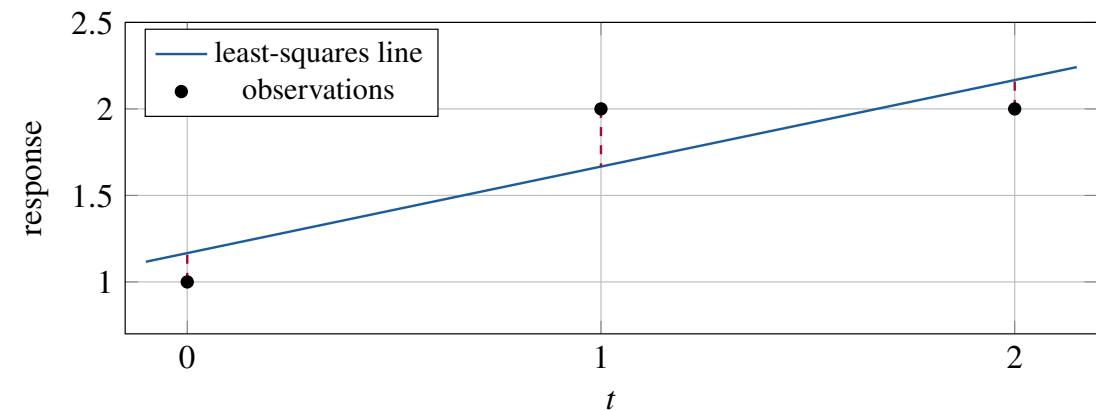
Example: check the fitted values and residuals

Residuals are signed vertical distances. A prediction is evaluation of the fitted function.

$$\hat{\mathbf{b}} = \begin{pmatrix} 7/6 \\ 5/3 \\ 13/6 \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} -1/6 \\ 1/3 \\ -1/6 \end{pmatrix}$$

$$\text{SSE} = \frac{1}{36} + \frac{1}{9} + \frac{1}{36} = \frac{1}{6}$$

$$(t_i, b_i) = (0, 1), (1, 2), (2, 2), \quad A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$



Example: fit the three Lecture 5 measurements

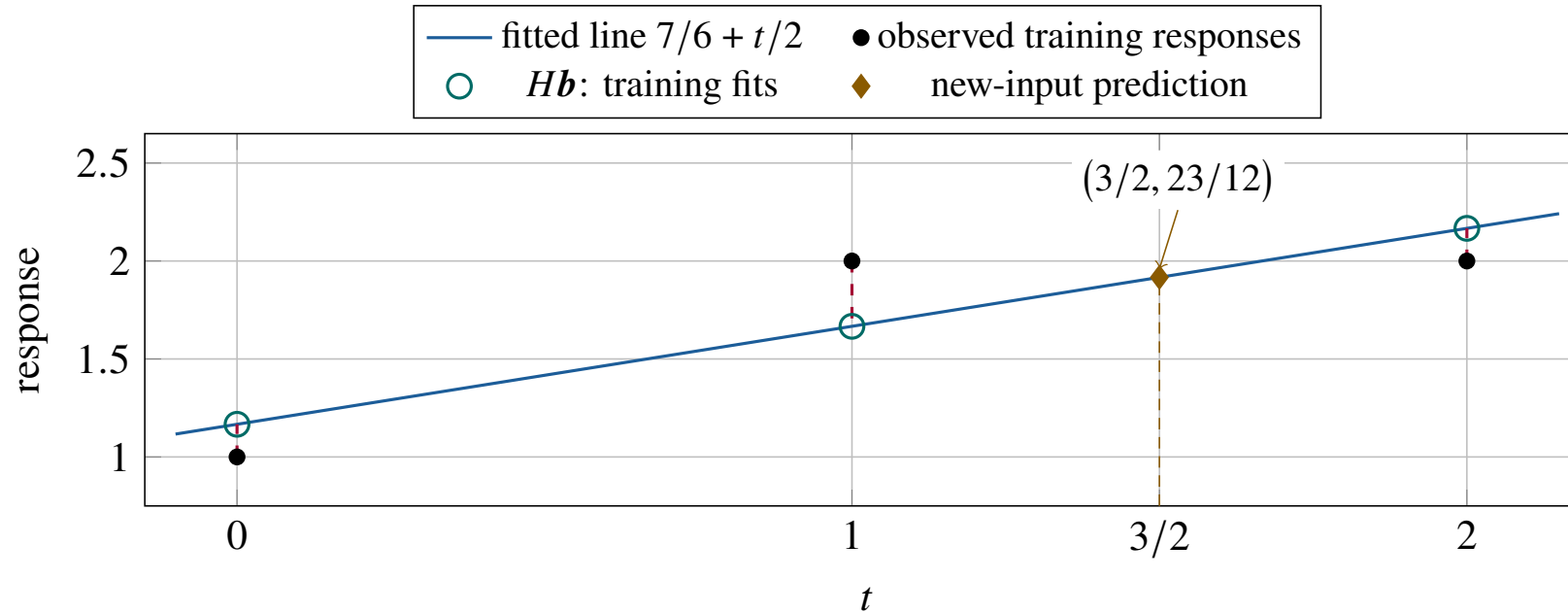
Build the design matrix, then solve the two normal equations.

$$(t_i, b_i) = (0,1), (1,2), (2,2), \quad A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

$$2\beta = 1, \quad \beta = \frac{1}{2}, \quad \alpha = \frac{7}{6}, \quad \hat{f}(t) = \frac{7}{6} + \frac{1}{2}t$$

Example: fitted values and a new prediction



Evaluate the fitted line at a new input; no new response is needed.

Analytic solution for fitting a line

Assume at least two distinct inputs. Center the inputs and responses. The fitted line passes through their mean point.

$$\bar{t} = \frac{1}{m} \sum_i t_i, \quad \bar{b} = \frac{1}{m} \sum_i b_i$$

$$\beta = \frac{\sum_i (t_i - \bar{t})(b_i - \bar{b})}{\sum_i (t_i - \bar{t})^2}, \quad \alpha = \bar{b} - \beta \bar{t}$$

Example 2

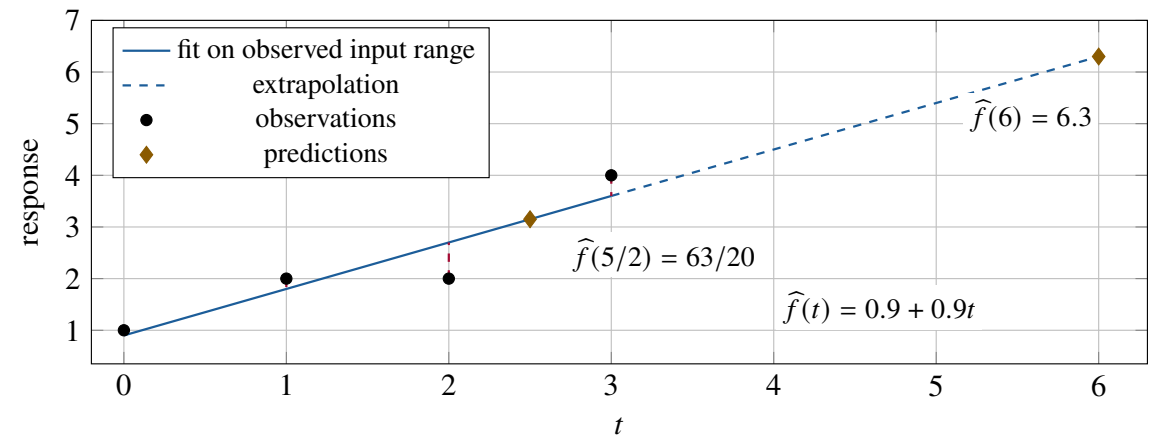
Use the centered-data formula to compute slope and intercept.

$$\mathbf{t} = (0, 1, 2, 3)^\top, \quad \mathbf{b} = (1, 2, 2, 4)^\top$$

$$\hat{f}(t) = \frac{9}{10} + \frac{9}{10}t, \quad \mathbf{r} = \frac{1}{10}(1, 2, -7, 4)^\top$$

$$\text{SSE} = \frac{1 + 4 + 49 + 16}{100} = \frac{7}{10}$$

$$\hat{f}\left(\frac{5}{2}\right) = \frac{63}{20} = 3.15, \quad \hat{f}(6) = 6.3$$



Polynomial regression

A curved model can remain linear in its unknown coefficients.

$$f_d(t) = x_0 + x_1 t + \cdots + x_d t^d$$

$$A_d = \begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^d \\ 1 & t_2 & t_2^2 & \cdots & t_2^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^d \end{pmatrix}, \quad \mathbf{x}_* = A_d^+ \mathbf{b}$$

Theorem 3: unique polynomial coefficients

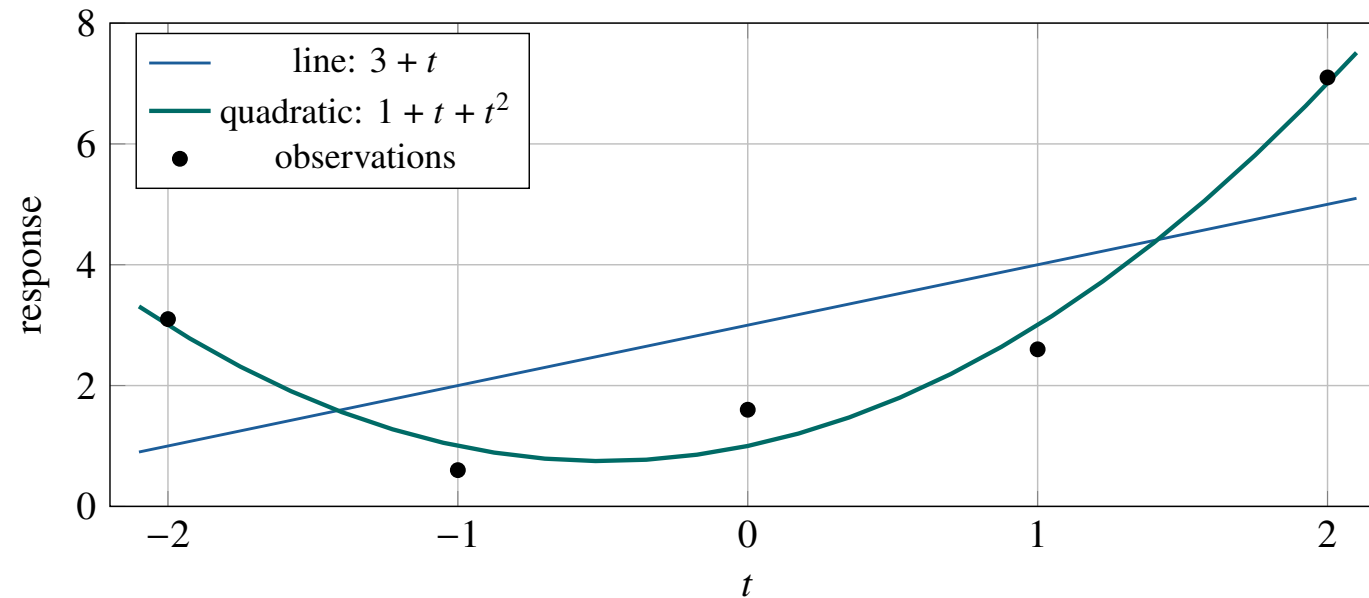
The number of distinct inputs determines whether the coefficients are unique.

$$\text{rank}(A_d) = d + 1 \iff \text{at least } d + 1 \text{ distinct inputs}$$

model	distinct inputs needed
line	2
quadratic	3
quartic	5

Example: polynomial regression

t_i	-2	-1	0	1	2
b_i	3.1	0.6	1.6	2.6	7.1



$$\hat{f}_1(t) = 3 + t$$

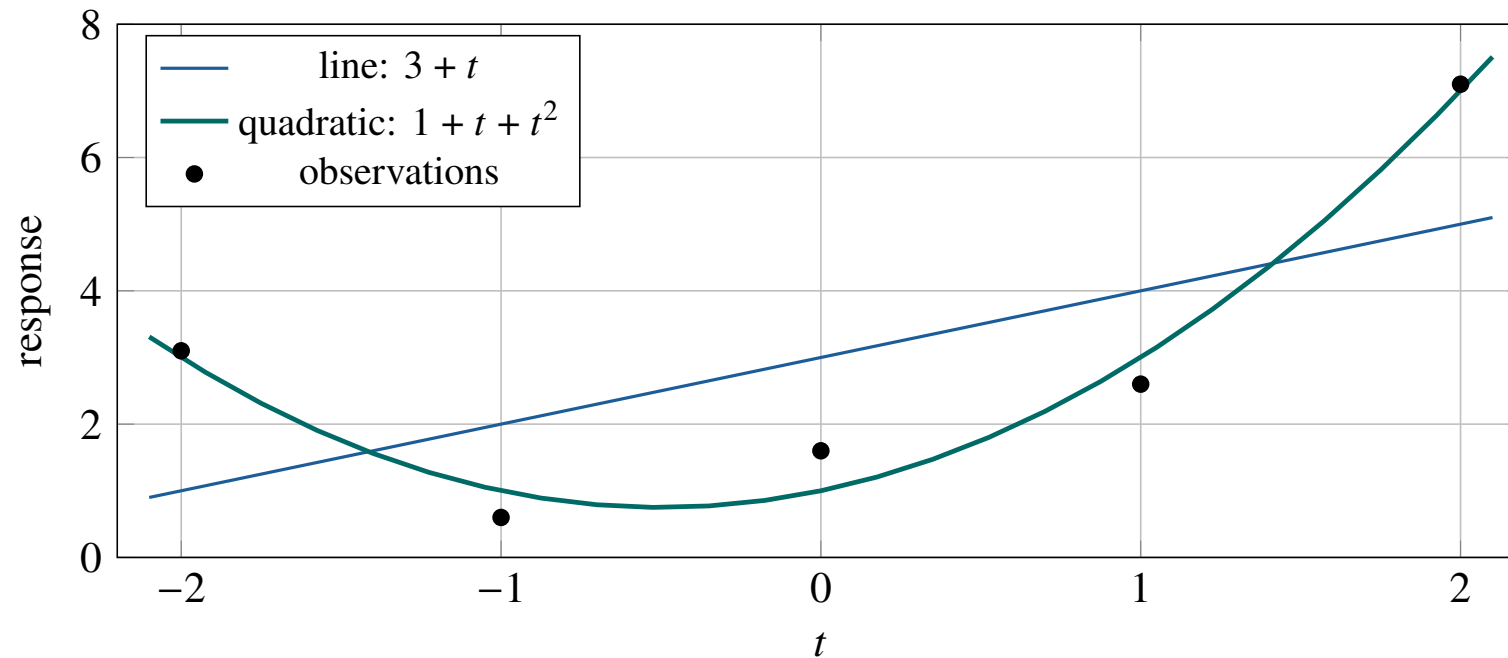
The quadratic design matrix

The quadratic adds a column for the squared input.

$$A_2 = \begin{pmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 3.1 \\ 0.6 \\ 1.6 \\ 2.6 \\ 7.1 \end{pmatrix}$$

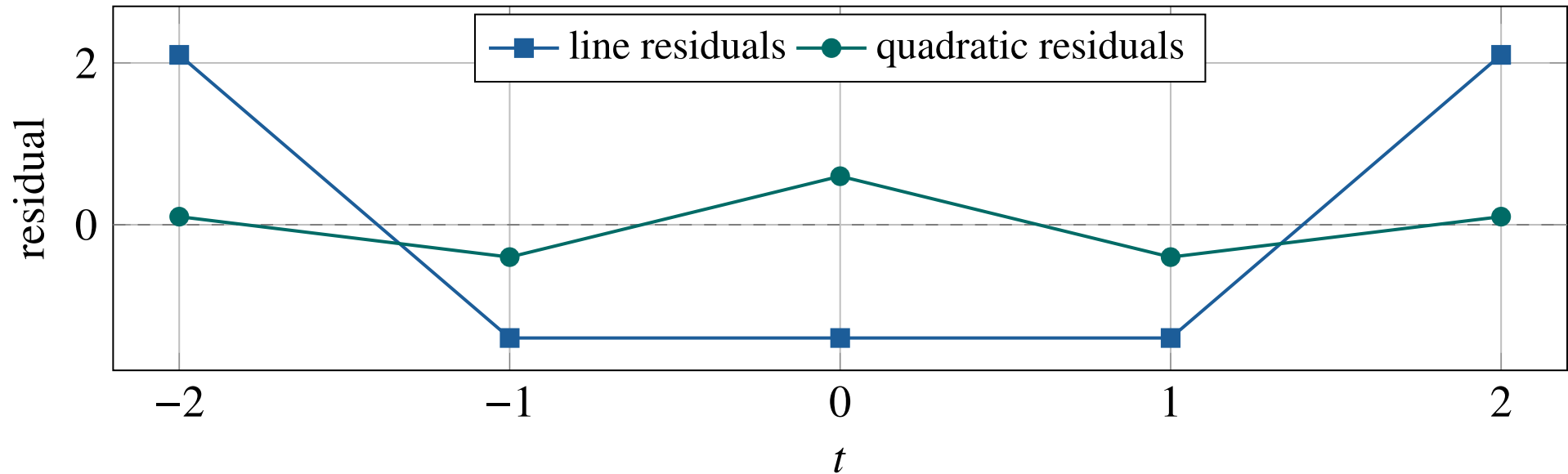
$$A_2^\top A_2 = \begin{pmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{pmatrix}, \quad A_2^\top \mathbf{b} = \begin{pmatrix} 15 \\ 10 \\ 44 \end{pmatrix}$$

Solving for the quadratic



$$\hat{f}_2(t) = 1 + t + t^2$$

Example 6: reading the residuals



Underfitting: the model is too restricted to describe the relevant relationship.

Overfitting: the fit follows training variation that predicts new responses poorly.

Making predictions on unseen data

Fit on the training set T . Compare every candidate on the same reserved validation set V .

$$\text{MSE}_{\text{train}}(d) = \frac{1}{|T|} \sum_{i \in T} \left[b_i - \hat{f}_{d,T}(t_i) \right]^2$$

$$\text{MSE}_{\text{val}}(d) = \frac{1}{|V|} \sum_{i \in V} \left[b_i - \hat{f}_{d,T}(t_i) \right]^2$$

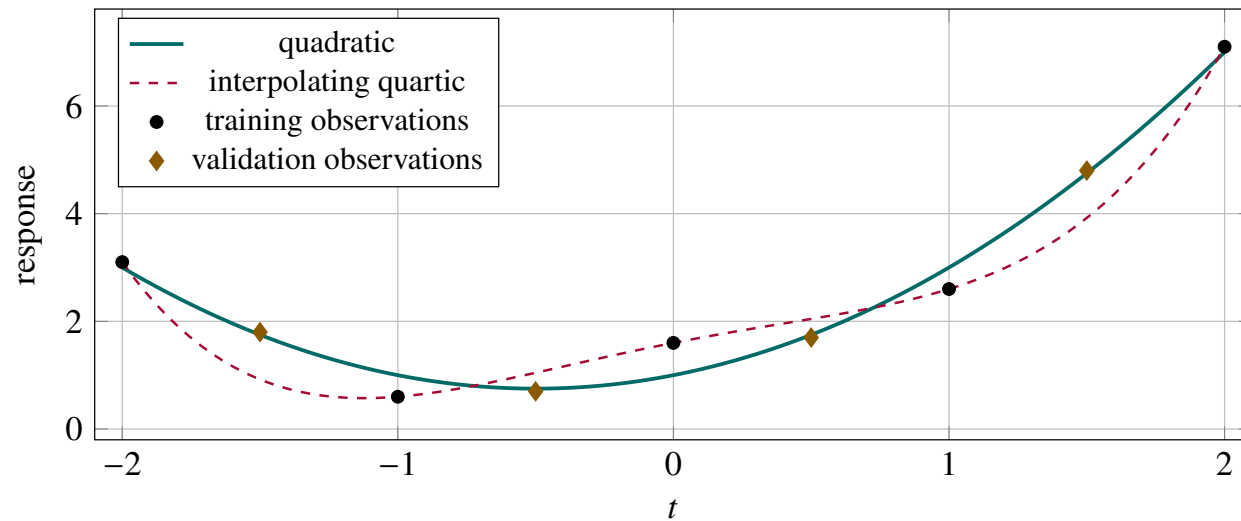
Both scores average squared prediction errors. The fitted coefficients stay fixed during validation.

Example: two separate prediction problems

training t	-2	-1	0	1	2
b	3.1	0.6	1.6	2.6	7.1
validation t	-1.5	-0.5	0.5	1.5	
b	1.80	0.70	1.70	4.80	

The quadratic predicts best here

d	training MSE	validation MSE
1	2.940	1.665000
2	0.140	0.002500
4	0	0.446545



Least squares as supervised learning

Supervised observations pair inputs with numerical target responses.

$$\boldsymbol{\phi}(t) = (\phi_0(t), \dots, \phi_{p-1}(t))^\top, \quad f_x(t) = \boldsymbol{\phi}(t)^\top \mathbf{x}$$

$$\ell(b, \hat{b}) = (b - \hat{b})^2$$

$$\widehat{R}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \ell(b_i, f_x(t_i)) = \frac{1}{m} \|\mathbf{b} - A\mathbf{x}\|_2^2$$

Features must be available when making the intended prediction.

Learned neural features

A neural network can learn its features as well as their final combination.

$$f_{\theta}(t) = c + \sum_{j=1}^q a_j \sigma(w_j t + v_j)$$

$$\theta = (c, a_1, \dots, a_q, w_1, \dots, w_q, v_1, \dots, v_q)^{\top}$$

$$\sigma(s) = \tanh(s) \quad \text{or} \quad \sigma(s) = \max(0, s)$$

Changing w_j or v_j changes the feature function $\sigma(w_j t + v_j)$.

So now the optimization problem becomes non-linear! We generally optimize using gradient descent.

A frozen hidden layer gives a least-squares problem

Fix the hidden weights w_j and biases v_j . Fit only the linear output layer.

$$\Phi_{i,:} = (1, \sigma(w_1 t_i + v_1), \dots, \sigma(w_q t_i + v_q))$$

$$\boldsymbol{\beta} = (c, a_1, \dots, a_q)^\top$$

$$\min_{\boldsymbol{\beta}} \|\mathbf{b} - \Phi\boldsymbol{\beta}\|_2^2, \quad \boldsymbol{\beta}_* = \Phi^+ \mathbf{b}$$

If the hidden weights also change, the design matrix changes during training.